

THEORETICAL AND COMPUTATIONAL ASPECTS OF INTEGER  
MULTICOMMODITY NETWORK FLOW PROBLEMS

A THESIS

Presented to

The Faculty of the Division of Graduate  
Studies and Research

By

James Robert Evans

In Partial Fulfillment  
of the Requirements for the Degree  
Doctor of Philosophy  
in the School of Industrial and Systems Engineering

Georgia Institute of Technology

May, 1975

THEORETICAL AND COMPUTATIONAL ASPECTS OF INTEGER  
MULTICOMMODITY NETWORK FLOW PROBLEMS

Approved:

John J. Jarvis, Chairman

C. M. Shetty

V. E. Unger

R. L. Rardin

R. A. Duke

Date approved by Chairman: 4/25/75



## ACKNOWLEDGMENTS

Education is a dynamic process. My deepest gratitude goes to Dr. John J. Jarvis, my academic advisor and chairman of my dissertation committee for his unending personal interest in my career and successful completion of my doctoral program. As a professional colleague and as a personal friend, he has enabled me to accomplish this goal and attain a greater sense of intellectual and educational maturity.

I also wish to thank Drs. R. L. Rardin, C. M. Shetty, V. E. Unger, and R. A. Duke for serving on my reading committee and contributing their knowledge to this work. My gratitude extends to Dr. R. N. Lehrer, Dr. W. W. Hines, and the entire Industrial and Systems Engineering faculty and staff for providing financial and moral support throughout my doctoral program.

I wish to express a special note of appreciation to Dr. Thomas W. Hill, Jr., who, while at Purdue University, introduced me to the fascinating subjects of graph theory and matroid theory; and to Dr. Don T. Phillips who "turned me on" to operations research.

Finally, I thank three very special people: my parents, Tony and Rita Evans, without whose love, encouragement, and financial support I would never have achieved this goal. This thesis is dedicated to them. And my gratitude goes to my fiancée, Beverly, whose love and encouragement helped me keep my spirits up whenever I found a counterexample to one of my numerous conjectures.

## TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS . . . . .	ii
LIST OF TABLES . . . . .	v
LIST OF ILLUSTRATIONS . . . . .	vi
SUMMARY. . . . .	viii
Chapter	
I. INTRODUCTION AND LITERATURE SURVEY . . . . .	1
Introduction	
Formulation of Multicommodity Network Flow Problems	
Theoretical Properties of Single and Multi-	
Commodity Flow Problems	
Feasibility Conditions and Two-Commodity Flows	
The General Multicommodity Flow Problem	
Applications of Multicommodity Networks	
Organization of the Dissertation	
II. MATROIDS, UNIMODULARITY, AND THE MULTICOMMODITY	
TRANSPORTATION PROBLEM . . . . .	19
Introduction	
Unimodular and Totally Unimodular Matrices	
Equivalent Linear Systems	
Matroid Theory, Graphs, and Unimodular Matrices	
Graphic Matroids and Linear Programming	
A Unimodularity Theorem for the Multicommodity	
Transportation Problem	
Matroid Graphs of Graphic MCTP Matroids	
An Equivalent Single Commodity Network for the	
MCTP(m,2,r)	
Proof of Theorem 2.13	
III. A GRAPH-THEORETIC CHARACTERIZATION OF INTEGER BASIC	
FEASIBLE SOLUTIONS IN MULTICOMMODITY NETWORKS . . . .	78
Introduction	
Single Commodity Bases	
Solution of Linear Equations by Change of Variables	
Non-unimodularity in Multicommodity Flow Bases	
Chain Groups and Cycles in Directed Graphs	

Chapter	Page
Chains and Cycles in Multicommodity Networks An Alternate Proof of the MCTP Unimodularity Theorem	
IV. COMBINATORIAL COMPLEXITY OF INTEGER MULTICOMMODITY TRANSPORTATION PROBLEMS . . . . .	103
Introduction	
The Multicommodity Assignment Problem	
Matroid Theory and Combinatorial Optimization	
Application to the Multicommodity Assignment and Transportation Problems	
V. SOLUTION STRATEGIES FOR THE INTEGER MULTICOMMODITY TRANSPORTATION PROBLEM . . . . .	118
Introduction	
Extended Matroid Graphs	
Some Apparent Theoretical Advantages of Extended Matroid Graphs	
A Heuristic Algorithm for the Integer MCTP	
VI. SOME TECHNIQUES APPLICABLE TO THE GENERAL INTEGER MULTICOMMODITY FLOW PROBLEM. . . . .	133
The General Integer Cycle Formulation	
A Combined Relaxation-Cutting Plane Strategy	
Other Solution Strategies	
VII. CONCLUSIONS AND RECOMMENDATIONS . . . . .	146
APPENDIX. TUTTE'S ALGORITHM FOR GRAPHIC MATROIDS . . . . .	152
BIBLIOGRAPHY . . . . .	159
VITA . . . . .	169

## LIST OF TABLES

Table	Page
1. Sensitivity Analysis for Arc Capacities . . . . .	127
2. Computational Results . . . . .	131

## LIST OF ILLUSTRATIONS

Figure	Page
1. A Multicommodity Network . . . . .	3
2. Multicommodity Transportation Network. . . . .	6
3. Three-Commodity Network . . . . .	8
4. Sakarovitch's Network . . . . .	9
5. Counterexample to Minimal Disconnecting Set Conjecture .	10
6. Rothchild and Whinston's Networks . . . . .	10
7. Production Planning Network . . . . .	15
8. Multi-Modal Transportation Example. . . . .	16
9. A Linear Graph, $G$ . . . . .	31
10. Dual Graph, $G^*$ . . . . .	33
11. Matroid Graph. . . . .	36
12. The Graph $G(2,2,2)$ . . . . .	63
13. The Graph $G^*(2,2,2)$ . . . . .	63
14. The Graph $G^*(2,2,r)$ . . . . .	65
15. The Graph $G(2,2,3)$ . . . . .	66
16. The Graph $G(3,2,2)$ . . . . .	66
17. The Graph $G'(2,2,2)$ . . . . .	68
18. The Graph $G'(3,2,2)$ . . . . .	74
19. Reduced Matroid Graph for MCTP(3,2,2). . . . .	75
20. A Linear Programming Network. . . . .	80
21. A Rooted Spanning Forest . . . . .	80
22. An MCTP(3,3,2) Network. . . . .	85

Figure	Page
23. Optimal Rooted Spanning Forests . . . . .	85
24. Optimal LP Solution . . . . .	87
25. A Two-Commodity Transportation Basis . . . . .	88
26. Cycle Flow Exchange for Commodity 1. . . . .	88
27. Cycle Flow Exchange for Commodity 2. . . . .	89
28. A Directed Graph . . . . .	94
29. Two Directed Cycles . . . . .	95
30. A Directed Graph . . . . .	96
31. A Two-Source, n-Sink Transportation Network . . . . .	99
32. An MCTP(3,3,4). . . . .	108
33. MCAP(3,3,2). . . . .	115
34. Independent Set in $M_1$ . . . . .	115
35. EMG(3,3,2) . . . . .	119

## SUMMARY

The nature of integer solutions in multicommodity network flow problems is investigated. A relationship between matroid theory, network flows, and integer programming is developed, and the results are applied to prove that a certain class of multicommodity transportation problems have totally unimodular constraint matrices and can be solved by equivalent single commodity network flow problems. A simple graph-theoretic condition is derived by which one can determine when a specific basis or all bases to a multicommodity flow problem will yield integer solutions. The combinatorial complexity of the integer multicommodity transportation problem is discussed, and a heuristic algorithm is developed. Applications of integer programming techniques to general multicommodity flow problems are also considered.

## CHAPTER I

### INTRODUCTION AND LITERATURE SURVEY

#### Introduction

Network flow theory has emerged as one of the most important areas of operations research from both a theoretical and applied standpoint since Ford and Fulkerson published their now classic paper [31]. Numerous applications are found in the areas of transportation, communication, distribution, project scheduling, inventory theory; indeed, in nearly every facet of operations research. Just recently, a new international journal, Networks, has appeared, devoted exclusively to the applications and theory of network flows.

Much of the theoretical literature in network flow theory falls naturally into two distinct classes: single commodity and multi-commodity flow problems. The distinction is well deserved since many properties of single commodity problems do not generalize to the multi-commodity case, for example, total unimodularity of the constraint matrix and the Max Flow-Min Cut Theorem. The theory and algorithms for single commodity flow problems are well developed; the contrary is true for multicommodity problems. While all single commodity network flow problems terminate integer, multicommodity problems, in general, do not. Since many practical applications of multicommodity networks are more easily interpreted with integer solutions, the author feels that this aspect of the problem merits attention.

A particular class of multicommodity flow problems, namely, the



multicommodity transportation problem, will be the central focus of this research, although many of the results will generalize to the general multicommodity flow problem.

The objectives of this study are:

- (i) to investigate theoretical properties of integer solutions to multicommodity network flow problems,
- (ii) to investigate the computational aspects of various approaches for solving integer multicommodity network flow problems, and
- (iii) to attempt to utilize topological properties of networks in solving the integer multicommodity flow problem.

#### Formulation of Multicommodity Network Flow Problems

Multicommodity networks are natural extensions of single commodity networks in which several distinct commodities flow through a network simultaneously over arcs which restrict the total amount of flow of all commodities. The objective is usually to either maximize the sum of all commodity flows between given sources and sinks, or to obtain the minimal cost routing of flows in a network to meet specified requirements. Networks may be directed, in which case the commodity may flow only in the direction of the arcs, or undirected, in which flow may occur in either direction. For example, consider the network in Figure 1. We will assume without loss of generality a single source and sink for each commodity, since a super source or super sink may be added to the network. Nodes  $s^k$  and  $t^k$  denote the source and sink, respectively, for commodity  $k$ . The network in Figure 1 is a directed

multicommodity network.

We may formulate the multicommodity maximum flow problem (MCMF) mathematically as follows:

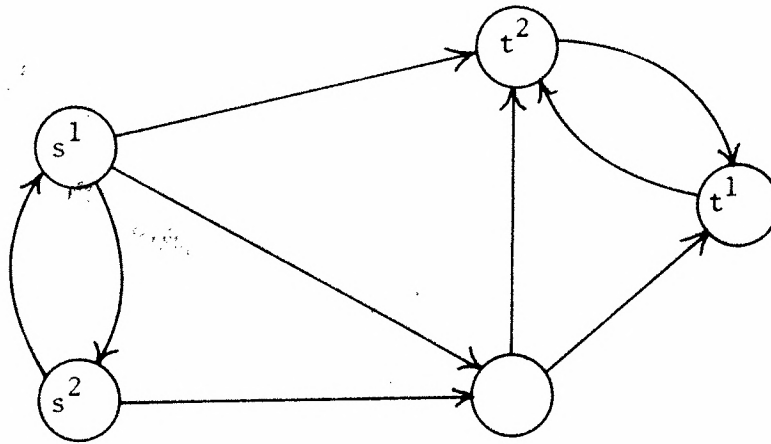


Figure 1. A Multicommodity Network

$$\text{MCMF-1:} \quad \text{Max } F = \sum_{k=1}^r v^k$$

$$\text{subject to} \quad \sum_{j \in N} f_{ij}^k - \sum_{j \in N} f_{ji}^k = \begin{cases} v^k & i = s^k \\ 0 & i \neq s^k, t^k \\ -v^k & i = t^k \end{cases}$$

$$\sum_{k=1}^r f_{ij}^k \leq u_{ij} \quad \text{for all } (i,j) \in A$$

$$f_{ij}^k \geq 0 \quad \text{for all } (i,j) \in A \text{ and } k$$

where  $N$  is the set of nodes of the network

$A$  is the set of arcs of the network

$f_{ij}^k$  is the flow of commodity  $k$  from node  $i$  to node  $j$

$v^k$  is the total flow of commodity  $k$  from  $s^k$  to  $t^k$   
 $u_{ij}$  is the capacity of arc  $(i,j)$   
 $s^k$  and  $t^k$  is the source and sink, respectively, for commodity  $k$ .

This formulation is called the node-arc formulation of the multicommodity maximum flow problem. An alternate formulation called the arc-path formulation can be described as follows

$$\begin{aligned} \text{MCMF-2:} \quad & \text{Max } F = \sum_{j=1}^q x_j \\ & \text{subject to } \sum_{j=1}^q a_{ij} x_j \leq u_i \quad \text{for all } i \\ & \quad \quad \quad x_j \geq 0 \quad \text{for all } j \end{aligned}$$

where  $P_1, P_2, \dots, P_q$  is an enumeration of all paths in the network from all sources to their respective sinks,

$$a_{ij} = \begin{cases} 1 & \text{if arc } i \in P_j \\ 0 & \text{otherwise} \end{cases}$$

$u_i$  is the capacity of arc  $i$

$x_j$  is the flow on path  $P_j$

Jarvis [61] has shown the equivalence of these two formulations.

The multicommodity minimum cost flow problem (MCMC) has a similar formulation. However, in this case,  $v^k$  is a specified constant or requirement in node-arc form.

MCMC:

$$\begin{aligned}
& \text{Min} \quad \sum_{k=1}^r \sum_{(i,j) \in A} c_{ij}^k f_{ij}^k \\
& \text{subject to} \quad \sum_{j \in N} f_{ij}^k - \sum_{j \in N} f_{ji}^k = \begin{cases} v^k & i = s^k \\ 0 & i \neq s^k, t^k \\ -v^k & i = t^k \end{cases} \\
& \quad \sum_{k=1}^r f_{ij}^k \leq u_{ij} \quad \text{for all } (i,j) \in A \\
& \quad f_{ij}^k \geq 0 \quad \text{for all } (i,j) \in A \text{ and } k
\end{aligned}$$

MCMC can also be formulated in arc-path form, and Tomlin [124] has demonstrated their equivalence.

A special case of the MCMC that will be of concern in this dissertation is the multicommodity transportation problem (MCTP). This is simply a generalization of the well-known Hitchcock problem to multiple commodities. The mathematical formulation is given below.

MCTP:

$$\begin{aligned}
& \text{Min} \quad \sum_{k=1}^r \sum_{j=1}^n \sum_{i=1}^m c_{ij}^k x_{ij}^k \\
& \text{subject to} \quad \sum_{j=1}^n x_{ij}^k = a_i^k \quad \text{for all } i, k \\
& \quad \sum_{i=1}^m x_{ij}^k = b_j^k \quad \text{for all } j, k \\
& \quad \sum_{k=1}^r x_{ij}^k \leq u_{ij} \quad \text{for all } i, j \\
& \quad x_{ij}^k \geq 0 \quad \text{for all } i, j, k
\end{aligned}$$

where  $x_{ij}^k$  is the flow of commodity  $k$  from source  $i$  to sink  $j$ ,

$a_i^k$  is the supply of commodity  $k$  at source  $i$

$b_j^k$  is the demand for commodity  $k$  at sink  $j$

$u_{ij}$  is the capacity of arc  $(i,j)$ ,  $u_{ij} < \infty$

We assume that  $\sum_{i=1}^m a_i^k = \sum_{j=1}^n b_j^k$  for all  $k$ .

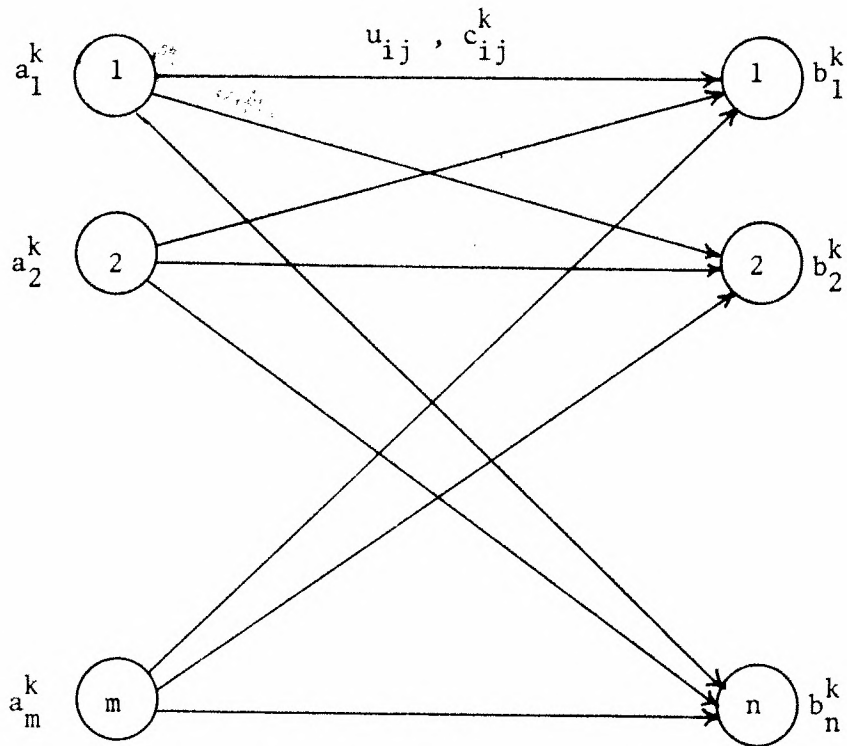


Figure 2. Multicommodity Transportation Network

### Theoretical Properties of Single and Multi-Commodity Flow Problems

For the maximal flow problem, two well-known properties that single commodity networks satisfy are (i) the maximum flow equals the value of the minimal cut set in the network, and (ii) if all capacities

are integer, the optimal flow values will be integer. Property (ii) holds also for the minimal cost problem if the requirements are also integer. This integrality property is a result of the fact that the constraint matrix in the node-arc formulation is totally unimodular. This means that every square submatrix has a determinant of  $\pm 1$  or 0. In particular, every basis has a determinant of  $\pm 1$ , and therefore simplex pivot operations will maintain integrality of the right hand side vector. As a result, simplex operations are essentially additions and subtractions, and network labelling algorithms have been devised to take advantage of this property. The most famous algorithms of this type are Ford and Fulkerson's maximum flow algorithm and the out-of-kilter algorithm [30].

For multicommodity flow problems, the constraint matrix is not generally totally unimodular, optimal solutions are not generally integer, and for maximal flow problems, the Max Flow-Min Cut Theorem does not necessarily hold. An example from Ford and Fulkerson [30, p. 17] will illustrate these facts. This network is shown in Figure 3. The maximum flow is  $9/2$ . This is accomplished by sending a flow of  $3/2$  from  $s^k$  to  $t^k$  for each  $k$ . In multicommodity networks, the analogous concept of a cut set is called a disconnecting set, and is defined as a set of arcs that break all paths from the sources to sinks of all commodities. In this example, the value of the minimal disconnecting set is 6. In general, the value of the minimal disconnecting set is greater than the maximum flow for multicommodity networks.

The capacity constraints, as we shall later see, destroy the total unimodularity of the constraint matrix in multicommodity

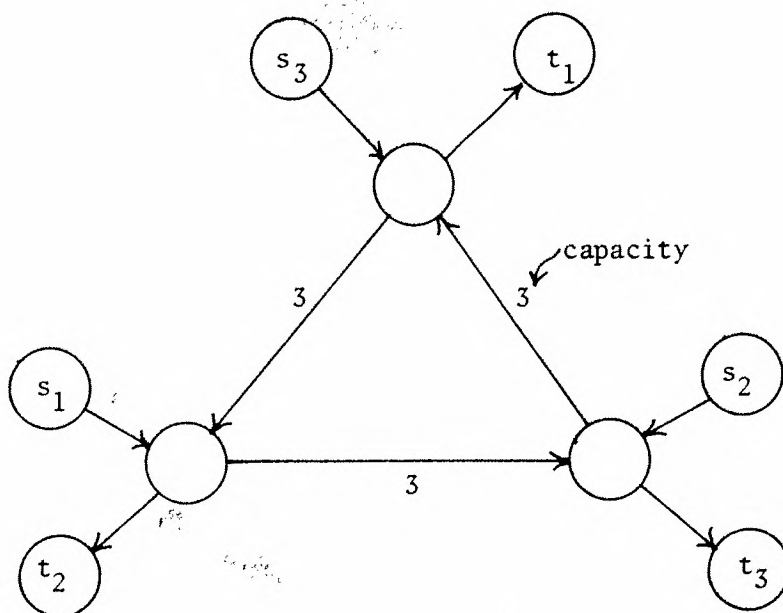


Figure 3. Three-Commodity Network

networks. Several attempts have been made to characterize the nature of rational solutions to multicommodity flow problems. For some cases, conditions are well established. In the two-commodity problem with undirected arcs, Hu [57] has shown that optimal solutions are multiples of 0,  $\pm 1$ , and  $\pm 1/2$  times the given data, that is, the maximum absolute value of any basis determinant is 2. However, Sakarovitch [115] gives an example whereby a multicommodity flow problem may assume any rational solution. This network is shown in Figure 4, and provided a counterexample to one of Jewell's conjectures [67]. In general, the nature of solutions to multicommodity flow problems remains unsolved.

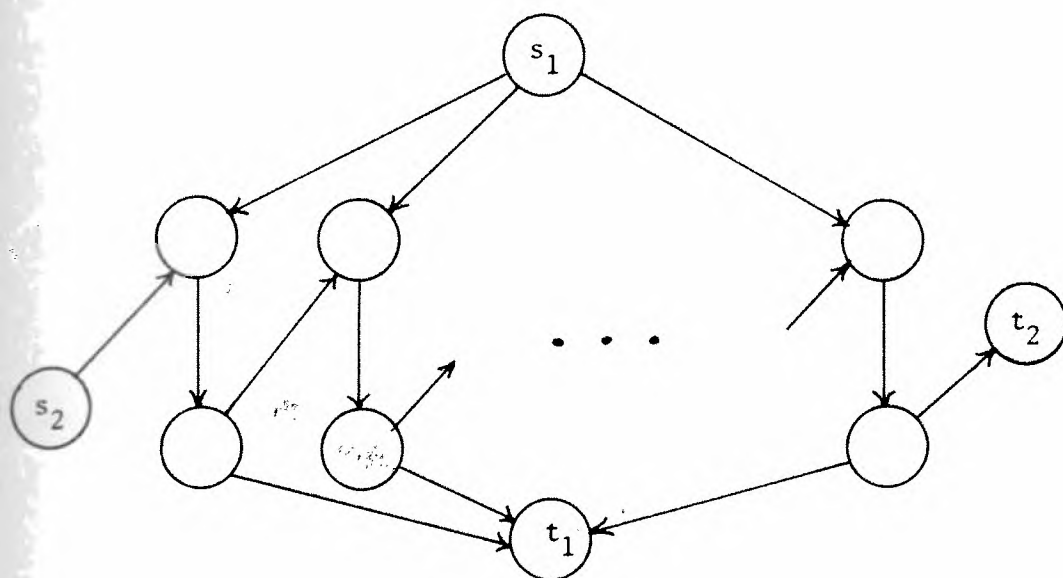


Figure 4. Sakarovitch's Network (all capacities are 1)

Several researchers have investigated extensions of the Max Flow-Min Cut Theorem to the multicommodity case. Many cases are related to two-commodity flows and are considered in the next section. One might conjecture that a minimal disconnecting set is the union of  $r$  single commodity minimum cuts (where  $r$  is the number of commodities). This is not the case, and Robacker [101] provided a counterexample (Figure 5) and proved that a minimal disconnecting set is a union of  $r$  single commodity cuts, which are, individually, not necessarily minimal.

Rothschild and Whinston [109] establish max flow-min cut conditions for certain special network structures; an  $r$ -commodity network which consists of a single line (Figure 6a), a two-commodity network in which each node is either a source or a sink and the network is a



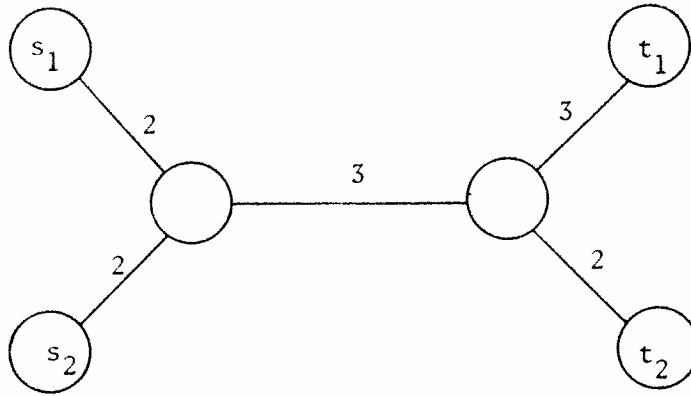


Figure 5. Counterexample to Minimal Disconnecting Set Conjecture

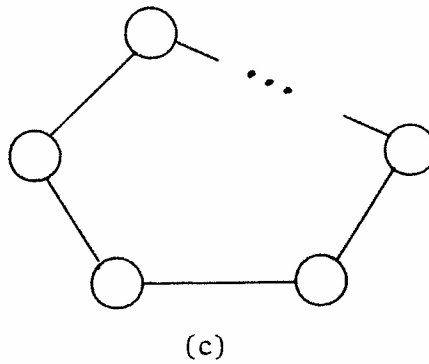
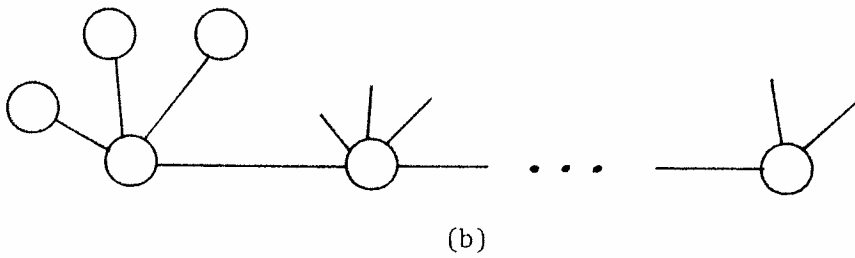
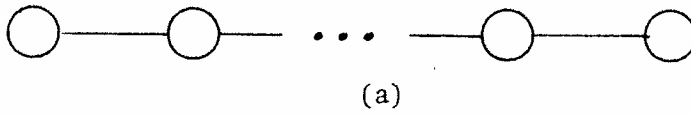


Figure 6. Rothchild and Whinston's Networks

tree (Figure 6b), and a two-commodity network which is a circuit (Figure 6c). The authors provide counterexamples to show that the last two types of networks do not generalize to an arbitrary number of commodities. One special result that holds for an arbitrary number of commodities is due to Kleitman, Martin-Löf, Rothschild, and Whinston [76]. If an undirected network has the property that every node is a source or sink for at least  $r - 1$  commodities, then max flow equals min cut. Rothfarb and Frisch [105] have shown that a three-commodity, undirected network with six nodes, each of which is a source or sink for some commodity has the max flow-min cut property.

In summary, most of the results of max flow-min cut theory are restricted to either two or three commodities, or to very special network structures, and provide little aid in attacking the general problem.

### Feasibility Conditions and Two-Commodity Flows

Much of the early research on the multicommodity flow problem is centered around two-commodity flows in undirected networks. One question considers the problem of finding maximum flows and the other concerns the question of feasibility: given a capacitated network and a set of  $r$  requirements, is it possible to construct a set of flows satisfying the requirements and not violating capacity restrictions, and if so, will the solution be integer?

The pioneering work in this area is Hu [57]. Hu considered both the feasibility problem and the maximum flow problem for a two-commodity undirected network. The question of feasibility for a general number of commodities has been subject to several conjectures

but remains unsolved.

Rothschild and Whinston [111] generalize Hu's work by providing a max flow-min cut theorem for two-commodity undirected networks in which every node is even; that is, the sum of the capacities of all arcs incident to a node is even. Such a network is called an Euler network by the authors. The constructive proof of this result leads to an algorithm which is discussed in [112]. A further paper [107] generalizes this result for Euler networks to answer the question of feasibility. This result is the same as Hu's for even capacities and requirements but only requires the assumption of even nodes.

#### The General Multicommodity Flow Problem

Historically, the arc-path formulation and a column generation algorithm was the first suggested approach to the multicommodity maximum flow problem [29]. This work later led to the development of the Dantzig-Wolfe Decomposition Principle [17]. Chen and DeWald [13] developed a labelling procedure based on the initial concepts in [29], and Tomlin [124] has extended the column generation idea to the minimum cost flow problem. Jewell [66] developed a primal-dual algorithm in 1958 and Sakarovitch [115] solved MCMF using a labelling process which required a linear system of equations to be solved at each iteration. Grinold [40] equated the problem to a polyhedral game. His algorithm, however, has slow convergence properties. Saigal [114] considered the problem in an "arc-circuit" formulation employing the property that a feasible solution may be viewed as a set of flows on spanning trees and a set of cycles in the network. This approach is closely related to Ellis Johnson's work on single commodity flows [70]. Hartman and

Lasdon [45] use these concepts in a generalized upper bounding framework.

More recently, Grigoriadis and White [39] have developed an algorithm based on Rosen's [103] primal partitioning technique. Their algorithm is a variation of the dual simplex method. Bazaraa [4] has specialized Balas' [2] infeasibility pricing method to the MCMF problem. Held, Wolfe, and Crowder [46] approach the problem from the context of Lagrangian duality [5, 26]. A survey of various methods for solving multicommodity flow problems with an emphasis on computational results can be found in Kennington [74].

The only known work which has investigated integer multicommodity flows is Bozoki [9], who considers the MCTP. His approach consists in solving the uncapacitated individual transportation problems and attempting to attain feasibility with the least increase in cost by rerouting the flow on oversaturated arcs. His algorithm is only a heuristic and no computational experience is reported.

Sakarovitch [115] has developed a condition for the maximal flow problem whereby the optimal solution will be integer-valued and gapless in the sense of the values of the maximum flow and minimal disconnecting set. The condition is that the network be "completely planar." A graph is said to be planar if it can be drawn on the plane in such a way that no edges intersect. A multicommodity network,  $G$ , is completely planar if

- (i) the graph obtained from  $G$  by linking a super source to the sources of all commodities and linking the sinks to a super sink is planar, and

- (ii) the graph obtained from  $G$  by adding the arcs  $(t^1, s^1), \dots, (t^r, s^r)$  is planar.

The constructive proof yields an algorithm, but the condition applies only to a very restricted class of networks.

In the current state-of-the-art there has been no serious attempt at characterizing the nature of integer solutions to multicommodity network flow problems, particularly, an analysis from a graph-theoretic point of view, and to utilize any results in algorithmic development.

### Applications of Multicommodity Networks

In this section we wish to discuss several, and by no means all, applications of multicommodity networks.

An obvious application is in the area of logistics and distribution. If several distinct commodities were shipped from several factories to regional warehouses by a common carrier with limited capacity per unit time, the problem is a multicommodity transportation problem. A more general problem is the transfer of empty railroad cars from various points of supply to areas where they are demanded. Each car type, e.g., boxcar, flatcar, refrigerated car, can represent a commodity. The regulated railroad schedules determine a capacity per unit time on a length of track. It is important to note that such a problem in reality is dynamic, not static. However, solution procedures for the static problem often aid in solving the dynamic problem.

Consider the problem of planning the production of several distinct products on common facilities over a finite horizon with deterministic demands. Figure 7 exhibits such a situation. An arc from node  $s$  to node  $j$  represents production in period  $j$ . The capacity

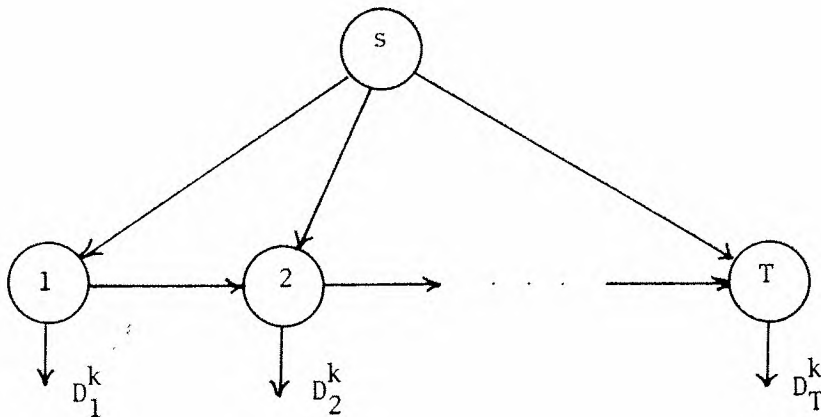


Figure 7. Production Planning Network

is limited by the facilities and workforce. An arc from node  $i$  to node  $j$  represents inventory carryover. A capacity constraint may represent available warehouse space.

A military communications system handles messages between commanders and their subordinates and is utilized by several branches of service (commodities). Switching capacity is available at the nodes for routing purposes. Given such a network and a set of requirements, one might ask how the messages can best be routed in the network so as to leave as much capacity available as possible.

Keith [73] has investigated the solution of a problem called a minimal closure problem. Several different types of commodities (men, materials, etc.) are located at various points in the world. A military situation arises which requires movement of commodities to certain ports, loading onto ships, and transportation to the objective area so that all ships arrive at the objective area in the minimal amount of time. This is a more complex multicommodity flow problem in which the



objective function is to minimize the maximum time that any ship travels to the objective area. Such problems are commonly called bottleneck problems. The two interacting components of this problem are first, the movement of the commodities to ports, and second, moving the ships to the ports and then to their ultimate destination.

Often in logistics problems, one has the capability of using several different modes of transportation, for instance, rail, truck, air freight. The commodities may transfer modes at various cities. This may be formulated as a multicommodity transportation problem, Figure 8. Each source node represents the arrival of the commodities via a particular carrier, and each sink represents the departure mode. Arc costs would represent labor transfer costs and capacities may represent a time constraint in making the transfer. Such a network would most likely represent only one node or city in an overall distribution problem.

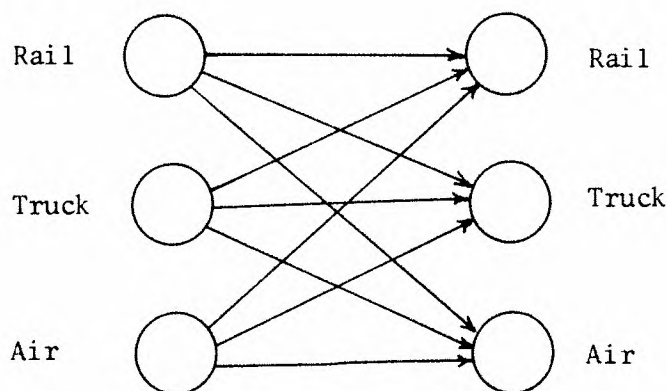


Figure 8. Multi-modal Transportation Example

These are but a few of the many problems that can be formulated as multicommodity network flow problems. There are many more in the areas of airline scheduling, freeway design, telephone and computer networks, etc.

### Organization of the Dissertation

In Chapter II we will present the major results of this research, that of relating matroid theory and unimodularity to the multicommodity transportation problem. These results will lay the foundation for some of the discussion in Chapter V.

Chapter III presents some new results concerning integer basic solutions to general multicommodity network flow problems. One result is used to provide an alternate proof of the main theorem in Chapter II.

In Chapter IV, matroid theory is revisited from a combinatorial optimization viewpoint and a relationship between the integer MCTP and the travelling salesman problem is developed. The multicommodity assignment problem is introduced and an extreme point property is proven.

Chapters V and VI are devoted to discussing solution techniques for integer multicommodity network flow problems. Chapter V relies on material developed in Chapter II and is restricted to the MCTP. Applications to the continuous MCTP are also discussed. Chapter VI discusses standard integer programming techniques applied to multicommodity networks.

Conclusions and recommendations for further research are presented in Chapter VII.



Theorems quoted in this thesis from other works may have been reworded slightly for greater clarity.

## CHAPTER II

### MATROIDS, UNIMODULARITY, AND THE MULTICOMMODITY TRANSPORTATION PROBLEM

#### Introduction

The purpose of this chapter is to develop the necessary background and theory to prove that a class of multicommodity transportation problems have totally unimodular constraint matrices. The constructive method of proof relies on matroid theory, and enables one to solve this class of MCTP's by single commodity network flow problems, exhibiting a new and unique application of matroid theory in the field of integer programming and network flows.

#### Unimodular and Totally Unimodular Matrices

In this section we wish to present a discussion of unimodularity and related concepts, and present some of the more important theorems concerning unimodularity in matrices. Consider a polyhedron  $P$  in  $n$ -dimensional Euclidean space. A polyhedron  $P$  has the integral property if every vertex of  $P$  has all integral coordinates. A matrix  $A$  is said to be unimodular if every basis has determinant  $+1$  or  $-1$ . A matrix  $A$  is said to be totally unimodular if the determinant of every square submatrix is  $+1$ ,  $-1$ , or  $0$ . Necessarily, each element of a totally unimodular matrix must be  $+1$ ,  $-1$ , or  $0$ . Every totally unimodular matrix is unimodular though not necessarily conversely. For instance, the following matrix is unimodular but not totally.

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 1 & 1 \end{bmatrix}$$

Let  $A$  be an  $m \times n$  matrix of integers,  $\underline{b}$  and  $\underline{b}'$  be  $m$ -tuples,  $\underline{d}$  and  $\underline{d}'$  be  $n$ -tuples. The components of  $\underline{b}$ ,  $\underline{b}'$ ,  $\underline{d}$ ,  $\underline{d}'$  are integers or  $\pm\infty$ . Let  $A_i$  denote the  $i$ th row of  $A$ , and  $A^j$  denote the  $j$ th column of  $A$ . Define the following polyhedra in  $n$ -space.

$$P(\underline{b}; \underline{b}') = \{\underline{x} | \underline{b} \leq A\underline{x} \leq \underline{b}'\}$$

$$Q(\underline{b}; \underline{b}'; \underline{d}; \underline{d}') = \{\underline{x} | \underline{b} \leq A\underline{x} \leq \underline{b}', \underline{d} \leq \underline{x} \leq \underline{d}'\}$$

If  $S$  is any set of  $k$  rows of  $A$  define the greatest common divisor of  $S$ ,  $\gcd(S)$  as follows

$$\gcd(S) = \begin{cases} 0 & \text{if every } k \times k \text{ determinant of } S \text{ equals } 0 \\ p & \text{where } p \text{ is the greatest common divisor of all } k \times k \text{ determinants of } S, \text{ otherwise} \end{cases}$$

Theorem 2.1 (Hoffman and Kruskal [55])

The following conditions are equivalent:

- (i)  $P(\underline{b}; \underline{b}')$  has the integral property for every  $\underline{b}, \underline{b}'$
- (ii)  $P(\underline{b}; \infty)$  has the integral property for every  $\underline{b}$
- (iii)  $P(-\infty; \underline{b}')$  has the integral property for every  $\underline{b}'$
- (iv) If  $r$  is the rank of  $A$ , then for every set  $S$  of  $r$  linearly independent rows of  $A$ ,  $\gcd(S) = 1$
- (v) For every set  $S$  of rows of  $A$ ,  $\gcd(S) = 1$  or  $0$

Theorem 2.2 (Hoffman and Kruskal [55])

The following conditions are equivalent:

- (i)  $Q(\underline{b}; \underline{b}'; \underline{d}; \underline{d}')$  has the integral property for every  $\underline{b}$ ,  $\underline{b}'$ ,  $\underline{d}, \underline{d}'$
- (ii) For some  $-\infty < \underline{d} < \infty$ ,  $Q(\underline{b}; \infty; \underline{d}; \infty)$  has the integral property for every  $\underline{b}$
- (iii) For some  $-\infty < \underline{d} < \infty$ ,  $Q(-\infty; \underline{b}'; \underline{d}; \infty)$  has the integral property for every  $\underline{b}'$
- (iv) For some  $-\infty < \underline{d}' < \infty$ ,  $Q(\underline{b}; \infty; -\infty; \underline{d}')$  has the integral property for every  $\underline{b}$
- (v) For some  $-\infty < \underline{d}' < \infty$ ,  $Q(-\infty; \underline{b}'; -\infty; \underline{d}')$  has the integral property for every  $\underline{b}'$
- (vi) The matrix  $A$  is totally unimodular

Any linear program with a totally unimodular constraint matrix has an optimal basic feasible solution which is integral; in fact, every basic feasible solution is integral. In linear programming, Theorem 2.1 may be sufficient for every basic feasible solution to be integer. Consider

$$\begin{aligned} \min \quad & \underline{c} \underline{x} \\ & A \underline{x} \geq \underline{b} \\ & \underline{x} \geq \underline{0} \end{aligned}$$

An equivalent statement is

$$\begin{aligned} \min \quad & \underline{c} \underline{x} \\ & A' \underline{x} \geq \underline{b}' \end{aligned}$$

where

$$A' = \begin{bmatrix} A \\ I \end{bmatrix} \quad \underline{b}' = \begin{bmatrix} \underline{b} \\ \underline{0} \end{bmatrix}$$

Conditions (ii) and (iv) of Theorem 2.1 imply that if every basis of  $A'$  has a determinant of absolute value 1, then every vertex of the set

$$P' = \{\underline{x} \mid A'\underline{x} \geq \underline{b}'\}$$

has integer coordinates. Hence there exists a vector  $\underline{x}^* \in P'$  which is all integer and  $\underline{c} \underline{x}^* \leq \underline{c} \underline{x}$  for all  $\underline{x} \in P'$ . It is therefore sufficient that  $A$  be unimodular (not necessarily totally unimodular) in order that the linear program have an optimal integer solution. Theorem 2.2 applies to linear programs with bounded variables.

Attempts to characterize unimodular and totally unimodular matrices have been initiated by several authors. A very useful theorem is the following.

Theorem 2.3 (Heller and Tompkins [53])

Let  $A$  be an  $m \times n$  matrix whose rows can be partitioned into two disjoint sets,  $S_1$  and  $S_2$ , such that  $A$ ,  $S_1$ , and  $S_2$  have the following properties:

- (i) Every entry in  $A$  is either 0, +1, or -1
- (ii) Every column contains at most two non-zero entries
- (iii) If a column of  $A$  contains two non-zero entries and both have the same sign, then one is in  $S_1$  and one is in  $S_2$
- (iv) If a column of  $A$  contains two non-zero entries and they are of opposite sign, then both are in  $S_1$  or both are in  $S_2$

Then  $A$  is totally unimodular.

From this theorem we may easily show that the node-arc incidence matrix of a directed graph is totally unimodular. Since every column contains exactly one  $+1$  and one  $-1$ , let  $S_1$  be the rows of the incidence matrix itself and  $S_2 = \emptyset$ . This implies that single commodity network flow problems have optimal integer solutions.

A corollary to Theorem 2.3 is given in [55]:

Corollary 2.3.1 (Hoffman and Kruskal [55])

If  $A$  is the incidence matrix of vertices versus edges of an undirected graph  $G$ , then in order that  $A$  be totally unimodular, it is necessary and sufficient that  $G$  have no cycles with an odd number of vertices.

Sufficiency follows from the well-known fact that a graph is bipartite if and only if it has no odd cycles and Theorem 2.3. If the graph has an odd cycle, let  $A'$  be the submatrix contained in the rows and columns of  $A$  corresponding to the vertices and edges of the cycle. Then the determinant of  $A'$  is  $\pm 2$  and necessity follows. Hence, the incidence matrix of an undirected bipartite graph is totally unimodular.

We will write  $A_i \geq A_j$  to indicate that row  $A_i$  is component-wise greater than or equal to row  $A_j$ . The following theorem derives historically from the integrality of transportation-type problems.

Theorem 2.4 (Hoffman and Kruskal [55])

Suppose  $A$  is a matrix of 0's and 1's and suppose the rows of  $A$  can be partitioned into disjoint sets  $S_1$  and  $S_2$  with this property: if  $A_i$

and  $A_j$  are both in  $S_1$  or both in  $S_2$ , and if there is a column  $A^k$  in which both  $A_i$  and  $A_j$  have a 1, then either  $A_i \leq A_j$  or  $A_i \geq A_j$ . Then  $A$  is totally unimodular. This is illustrated by the following matrix:

$$\begin{array}{l} S_1 \left\{ \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \right. \\ S_2 \left\{ \begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix} \right. \end{array}$$

Suppose that two columns of 0's and 1's have the property that the portions of them between (in the inclusive sense) their lowest common 1 and the lower of their higher separate 1's are identical. That is, if columns  $A^i$  and  $A^j$  are the following

$$\begin{array}{l} \begin{array}{cc} A^i & A^j \end{array} \\ \begin{array}{l} \text{row } k \rightarrow \\ \text{row } \ell \rightarrow \\ \\ \text{row } m \rightarrow \end{array} \begin{bmatrix} 0 & 0 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & 0 \\ 0 & 1 \\ 1 & 1 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ 1 & 1 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} \end{array}$$

where  $\ell > k$ , and  $m$  is the largest row index in which both columns have a 1, then all elements from row  $\ell$  to row  $m$  are identical in both columns. Then these two columns are said to be in accord. From this is derived

the following.

Theorem 2.5 (Hoffman and Kruskal [55])

Suppose A is a matrix of 0's and 1's and suppose that the rows of A can be rearranged in such a way that every pair of columns is in accord. Then A is totally unimodular.

This situation corresponds to a graph in which every vertex has at most one predecessor. The columns of A may represent any directed paths in the graph.

The following matrix can be shown to satisfy Theorem 2.5.

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Heller [51] characterizes unimodular matrices in terms of unimodular linear transformations; that is, the group of transformations whose determinants have absolute value 1. Some of the principal theorems are given below.

Theorem 2.6 (Heller [51])

A matrix A is unimodular if and only if every two bases are related by a unimodular transformation.



Theorem 2.7 (Heller [51])

A is unimodular if and only if for every basis  $B \subset A$  and every column  $A^i \in A$ , the coordinates of  $A^i$  with respect to  $B$  are equal to +1, -1, or 0.

This theorem states that if a column of  $A$  is a linear combination of a maximal set of independent columns of  $A$ , then the coefficients in the linear combination are 0, +1, or -1. This is often called the Dantzig Property.

Theorem 2.8 (Heller [51])

If a linear transformation  $T$  on a vector space  $V$  of dimension  $n$  preserves the dimension of  $V$ , then  $A$  is unimodular if and only if  $T(A)$  is unimodular.

Theorem 2.9 (Heller [51])

The set of columns of  $A$  is unimodular if and only if there exists a non-singular linear transformation  $T$  such that  $T(A)$  is a totally unimodular set.

Chandrasekaran [12] has characterized totally unimodular matrices in terms of the greatest common division of multiples of the rows of  $A$ .

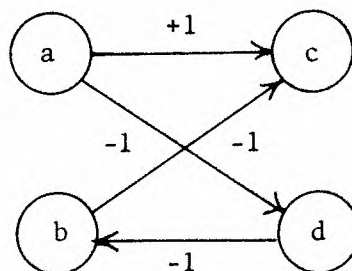
Theorem 2.10 (Chandrasekaran [12])

A matrix  $A$  is totally unimodular if and only if for every non-singular submatrix  $H = [h_{ij}]$  of  $A$ , the g.c.d. of  $\sum_j \lambda_j h_{1j}, \sum_j \lambda_j h_{2j}, \dots$ , is 1 for any  $\lambda_j = 0, +1, \text{ or } -1$  but not all 0.

Other characterizations of totally unimodular matrices are given by Cederbaum [11] and Camion [10]. Recently Commoner [14] has derived a rather simple sufficient condition for totally unimodular matrices. The condition is derived from concepts of regular abelian chain groups which will be discussed in the next chapter. Basically, Commoner's result is the following. Suppose we are given an arbitrary matrix of 0's, 1's, and -1's. We may construct a bipartite graph  $G$  by associating a vertex set  $M$  with the rows of  $A$  and a vertex set  $N$  with the columns of  $A$ . If  $a_{ij}$  is non-zero, construct a directed arc from  $i \in M$  to  $j \in N$  if  $a_{ij} = +1$ , or from  $j \in N$  to  $i \in M$  if  $a_{ij} = -1$ . Arbitrarily orient each cycle in  $G$ , and assign the value  $+1$  or  $-1$  to each arc in the cycle depending on whether it is oriented in the same or opposite direction as the cycle; i.e., a forward or reverse arc. The sign of the cycle is the product of these numbers taken over the arcs of the cycle. The result is that if every elementary cycle has sign  $+1$ , then  $A$  is totally unimodular. For example, let

$$A = \begin{matrix} & \begin{matrix} c & d \end{matrix} \\ \begin{matrix} a \\ b \end{matrix} & \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \end{matrix}$$

The corresponding graph is



with the orientation of the unique cycle being (a,c,b,d,a). The sign of the cycle is -1; hence A is not totally unimodular.

The major difficulty with any of these characterizations is implementation, particularly when trying to generalize to classes of matrices with similar structure. In the remainder of this chapter we shall discuss the relationship between matroid theory and unimodularity, and apply these results to the multicommodity transportation problem.

To motivate future discussions, we wish to quote from Rockafellar [102, p. 120]:

A typical way of proving that a given matrix A has the (total) unimodular property is to show that A can be constructed by a sequence of such operations from a matrix A' which in turn may be interpreted as a circulation matrix of some directed graph. Although A itself may no longer correspond directly to a directed graph, it does correspond to . . . (a regular matroid in the sense of Tutte). Linear programming manipulations of A therefore have graph-like interpretations which might be an important conceptual aid.

Phrases in parentheses were supplied by the author. The "operations" referred to above consist of elementary pivots on unit elements, taking submatrices, permutations of rows and columns, multiplying various rows and columns by -1, taking transposes, or appending a new row or column having only one non-zero component, and that a +1 or -1.

### Equivalent Linear Systems

Consider the two systems of equations

$$(I) \quad A\underline{x} = \underline{b}$$

$$(II) \quad A^*\underline{x} = \underline{b^*}$$

Systems (I) and (II) are equivalent if

$$\underline{x} \in P_1 \iff \underline{x} \in P_2$$

where

$$P_1 = \{\underline{x} \mid A\underline{x} = \underline{b}\}$$

$$P_2 = \{\underline{x} \mid A^*\underline{x} = \underline{b}^*\}$$

The usual pivot operations of linear programming preserve this equivalence. A sequence of pivot operations can be described by a linear transformation  $T$ ; that is, if  $A^* = T(A)$  and  $\underline{b}^* = T(\underline{b})$ , then (I) and (II) are equivalent, (provided  $T$  is of full rank).

Let  $A$  be a matrix of zeros and ones which has full row rank. Suppose there exists a linear transformation  $T : R^m \rightarrow R^m$  such that  $T(A) = A^*$  and  $A^*$  has the same rank as  $A$ . If  $A^*$  is totally unimodular, by Theorem 2.8,  $A$  is at least unimodular, though may not necessarily be totally unimodular. Further, suppose that every column of  $A^*$  has at most two non-zero entries, every non-zero entry is either +1 or -1, and if a column has two non-zero entries, then one is +1 and one is -1. By Theorem 2.3,  $A^*$  is totally unimodular. Moreover, we may obtain the node-arc incidence matrix of a directed graph  $G$ , denoted by  $N^*$ , by summing the rows of  $A^*$ , multiplying the result by -1, and appending this new row to  $A^*$ .

If (I) is the constraint system of a linear program, we have derived an equivalent linear program that is a pure network problem. Each variable  $x_j$  corresponds to an arc of  $G$ , and each row of  $A^*$

corresponds to conservation of flow at a node of  $G$ .

The advantage of doing this is that network programming problems are very easy to solve, and far easier than solving the problem by straightforward linear programming. In fact, Glover, Karney, Klingman, and Napier [37] have developed a network code for transportation problems that is approximately 150 times faster than the state-of-the-art linear programming codes. The disadvantage of this approach is that one may not know if such a transformation exists, or if it does, how to find it. In the remainder of this chapter we shall see that matroid theory may enable us to find this transformation quite easily.

### Matroid Theory, Graphs, and Unimodular Matrices

Whitney first introduced the concept of a matroid in 1935 as an abstraction of the notions of linear dependence and independence in vector spaces. In this section we wish to introduce the relevant concepts of matroid theory that apply directly to Rockafellar's statements in the previous section. For a more complete introduction, the interested reader is referred to Tutte's monograph [130] and Harary and Welsh [44]. Much of this section is taken from Tutte [130].

A matroid is a combinatorial structure defined on a finite set  $E$  along with a class  $Q$  of non-null subsets of  $E$ . The members of  $Q$  are called circuits of a matroid  $M$  if the following axioms hold.

Axiom C1    No member of  $Q$  is a proper subset of another

Axiom C2    Let  $a$  and  $b$  be distinct members of  $E$ . Let  $X$  and  $Y$  be members of  $Q$  such that  $a \in X \cap Y$  and  $b \in X - Y$ . Then there exists  $Z \in Q$  such that  $b \in Z \subseteq (X \cup Y) - \{a\}$

An equivalent statement of Axiom C2 is: If  $X$  and  $Y$  are members of  $\mathcal{A}$  and  $a \in X \cap Y$ , then  $(X \cup Y) - a$  contains a member of  $\mathcal{Q}$ . As an example of a matroid, let  $G$  be an undirected graph,  $E$  be the edge set of  $G$ , and  $\mathcal{Q}$  be the class of all cycles of  $G$ . In Figure 9, let  $X = \{a, b, e, d\}$  and  $Y = \{c, d, g, f\}$ . Then  $d \in X \cap Y$  and  $b \in X - Y$ . Also,  $(X \cup Y) - d = \{a, b, e, c, g, f\}$ . Let  $Z = \{a, b, c\} \subseteq \{a, b, e, c, g, f\}$ .  $Z$  then satisfies the condition of Axiom 2. One can show that the class of all proper cycles in a linear graph  $G$  is the class  $\mathcal{Q}$  of a matroid  $M$  defined on the edge set of  $G$ . We shall call this the circuit matroid of  $G$ . The reader may also verify that the set of proper cuts defines the circuits of a matroid in a linear graph. This will be called the bond matroid of  $G$ . As a third example, let  $A$  be an  $m \times n$  matrix of rank  $m$  ( $m < n$ ). Let  $\mathcal{Q}$  be the class of all sets of  $m + 1$  columns of  $A$ , where  $E$  is the set of columns of  $A$ . Then  $E$  and  $\mathcal{Q}$  satisfy axioms C1 and C2. There are several other equivalent axiom systems for matroids; one will be introduced in Chapter IV, and others are discussed in [44].

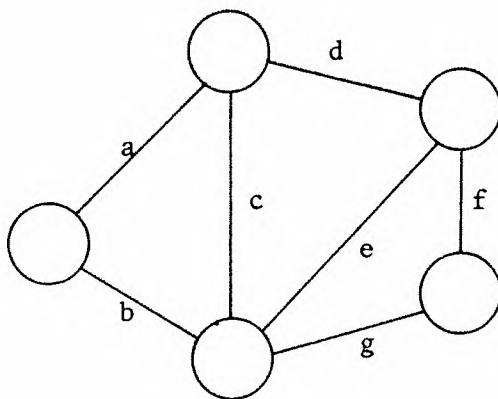
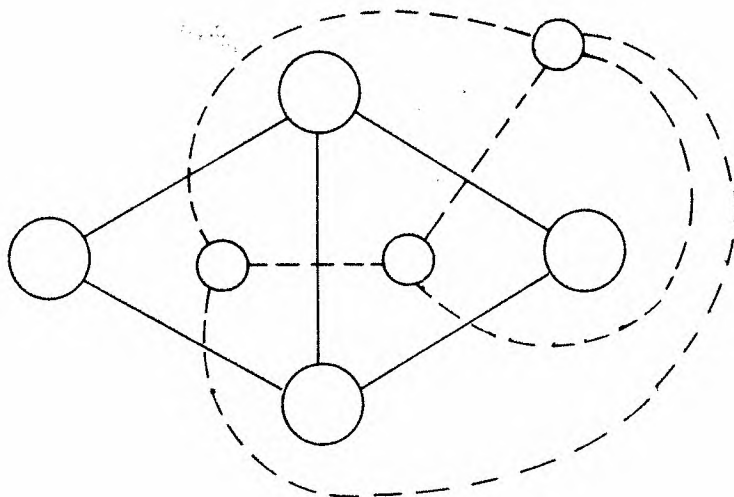


Figure 9. A Linear Graph,  $G$

Just as in linear programming, duality is fundamental in matroid theory. Every planar graph  $G$  has a (geometric) dual  $G^*$ .  $G^*$  is constructed by placing a vertex  $v_i^*$  in each region  $i$  of the plane that is partitioned by  $G$ . Corresponding to each edge  $e$  of  $G$ , we draw an edge  $e^*$  which crosses  $e$  (but no other edge in  $G$ ) and joins the vertices which lie in the two adjacent regions of  $e$ . For example,  $G$  is denoted by solid lines and  $G^*$  by dotted lines as shown below.

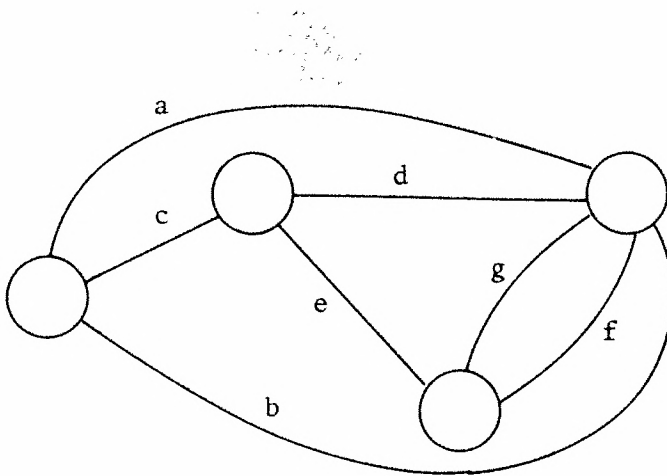


The dual of the graph in Figure 9 is given in Figure 10. Note that a cycle in  $G$  is a cut set in  $G^*$  and vice-versa. This is true of any pair of planar and dual graphs. From this idea arises the fact that the circuit matroid and bond matroid of a graph  $G$  are duals of each other.

A matroid is called graphic if it is the bond matroid of some finite graph, and cographic if it is the circuit matroid of a graph.

Let us first consider matrices composed of the elements 0 and 1. An  $m \times n$  matrix,  $R$ , of 0's and 1's can be thought of as a representative matrix of a matroid  $M$ . We will assume that the rank of  $R$  is  $m$



Figure 10. Dual Graph,  $G^*$ 

( $m < n$ ). If we choose  $m$  linearly independent columns of  $R$  and row reduce  $R$  modulo 2 so that  $R$  is of the form  $R_1$  where

$$R_1 = [I \ N]$$

we call  $R_1$  a standard representative matrix of  $M$ .

If  $M$  is a graphic matroid, then  $M$  is the bond matroid of a graph  $G$ , and every row of  $R_1$  is a bond (cut set) of  $G$ . Moreover, every column corresponding to  $N$  defines a cycle in  $G$ . For example, suppose  $R_1$  is the following

$$\begin{array}{c}
 \begin{array}{ccccccc}
 & a & c & e & f & g & b & d \\
 a & \left[ \begin{array}{ccccccc}
 1 & & & & & & 1 \\
 & 1 & & & & & 1 & 1 \\
 & & 1 & & 1 & & & 1 \\
 & & & 1 & 1 & & & 
 \end{array} \right]
 \end{array}
 \end{array}$$



Each row can be verified to be a cut set in Figure 9. Also, columns 5, 6, and 7 define the cycles efg, abc, and cde respectively. The graph G will be called a matroid graph.

Now let us consider matrices composed of 0's, 1's, and -1's. Such a matrix is called binary if replacement of -1's by 1's leaves the ranks of submatrices unaltered, where the rank of the derived matrix is with respect to modulo 2 arithmetic. The corresponding matroid is called a binary matroid. One important result is that every totally unimodular matrix is binary (Seshu and Reed [116, p. 110]). A matrix of integers mod 2 is called regular if the replacement of a suitable set of 1's by -1's makes it totally unimodular and leaves the ranks of the submatrices unaltered. An example of a matrix which is not regular is the following

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

A node-arc incidence matrix of an undirected graph is regular since we may replace a set of 1's by -1's to obtain the node-arc incidence matrix of a directed graph. This is equivalent to giving each edge an orientation. By Corollary 2.3.1, the mod 2 matrix need not be totally unimodular itself. A matroid is regular if its representative matrix is regular.

An important theorem relating graphic matroids and unimodularity is the following.

Theorem 2.11 (Tutte [132])

Every graphic matroid is regular.

This theorem is useful in the following sense. Given a matrix  $A$  of 0's and 1's, we may determine if its corresponding binary matroid is graphic. If so, we may construct an undirected graph  $G$  that topologically represents  $A$ , and secondly, we may replace a set of 1's by -1's to obtain the incidence matrix of a directed graph  $A^*$ . If  $A$  is the constraint matrix of a linear program, and this "replacement" is such that  $A^*$  can be obtained by a linear transformation on  $A$ , then we have an equivalent network problem. This is not always possible for all matrices that are graphic. The next section will discuss this.

The advantage of deriving the matroid graph is that it is often possible to determine the transformation  $T$  by labelling the arcs of the graph with the variables they represent and observing the structural relationships among the variables. In fact, this is the approach that will be used for the multicommodity transportation problem.

A more detailed development of the theoretical aspects of graphic matroids can be found in Tutte [130] and Seshu and Reed [116]. Rockafeller [102] provides some interesting comments relating Minty's [90] work on digraphoids to the subject.

We close this section by noting that Tutte has developed an algorithm for determining when a binary matroid is graphic [128]. A discussion of this algorithm is presented in the Appendix.

### Graphic Matroids and Linear Programming

Consider the following linear programming constraint matrix, A:

$$\begin{array}{cccccc}
 x_1 & x_2 & x_3 & x_4 & x_5 & x_5 & \underline{b} \\
 \left[ \begin{array}{cccccc}
 1 & 1 & 1 & 0 & 0 & 0 \\
 1 & 0 & 0 & 1 & 0 & 0 \\
 1 & 0 & 1 & 0 & 1 & 0 \\
 1 & 1 & 0 & 0 & 0 & 1
 \end{array} \right] & \left[ \begin{array}{c}
 b_1 \\
 b_2 \\
 b_3 \\
 b_4
 \end{array} \right]
 \end{array}$$

corresponding to the LP

$$\begin{aligned}
 \min \quad & \underline{c} \underline{x} \\
 & A\underline{x} = \underline{b} \\
 & \underline{x} \geq \underline{0}
 \end{aligned}$$

One can show that the matroid represented by A is graphic with matroid graph, G, given in Figure 11.

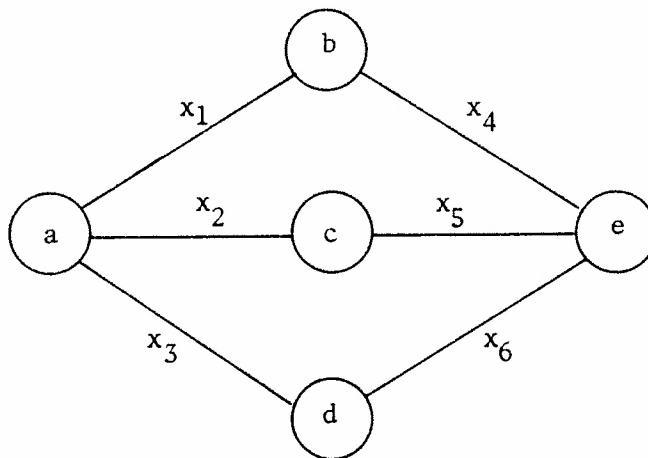


Figure 11. Matroid Graph

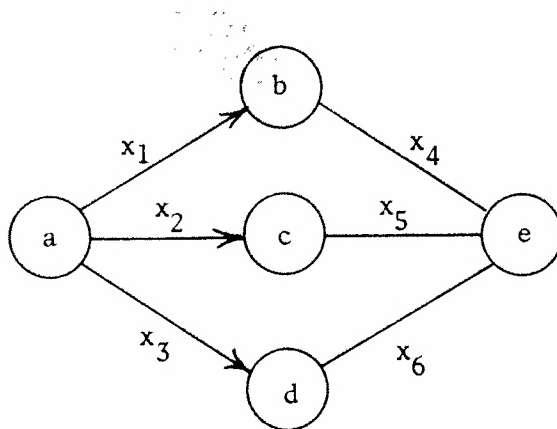
The node-arc incidence matrix,  $N$ , of  $G$  is given below.

$$\begin{array}{c}
 \\
 \\
 \\
 \\
 \\
 \end{array}
 \begin{array}{cccccc}
 x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\
 \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array}
 \left[ \begin{array}{cccccc}
 1 & 1 & 1 & 0 & 0 & 0 \\
 1 & 0 & 0 & 1 & 0 & 0 \\
 0 & 1 & 0 & 0 & 1 & 0 \\
 0 & 0 & 1 & 0 & 0 & 1 \\
 0 & 0 & 0 & 1 & 1 & 1
 \end{array} \right]
 \end{array}$$

Let  $N^-$  be the matrix consisting of the first four rows of  $N$ . (The rows of  $N^-$  are linearly independent mod 2, whereas  $N$  is not). If we can orient  $G$ , i.e., replace a set of 1's by -1 such that each column contains exactly one +1 and one -1 (denoted by  $N_d^-$ ), and show that  $N_d^-$  can be obtained from  $A$  by a sequence of elementary pivots, then the linear program can be solved as a pure network problem. We cannot simply arbitrarily assign -1's. However, knowing the topological structure of  $G$  and the relations between the  $x_j$ 's may help us to do this. For example, consider the conservation of flow constraint at node  $a$ , given by

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1'$$

where  $a_{ij} = +1$  or  $-1$ . If all the  $a_{ij}$  are +1 and  $b_1' = b_1$ , this is precisely the first row of  $A$ . We have now oriented the arcs  $x_1, x_2, x_3$ , as seen below.



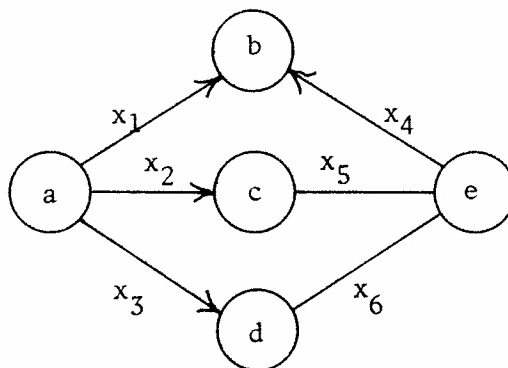
The constraint for node b is

$$a_{21}x_1 + a_{24}x_4 = b_2'$$

Since  $a_{11} = +1$ ,  $a_{21}$  must be  $-1$ . Also the form of the equation is the same as row 2 of A. Thus, by multiplying row 2 of A by  $-1$  we obtain

$$-x_1 - x_4 = -b_2$$

This establishes the orientation of  $x_4$ ,



The constraint at node c is

$$a_{32}x_2 + a_{35}x_5 = b_3'$$

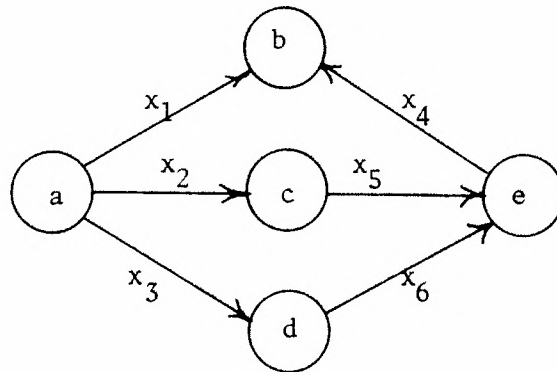
$a_{32}$  must be -1 since  $a_{12} = +1$ . To obtain an equivalent constraint, we need to find a linear combination of rows of A such that

$$-x_2 + a_{35}x_5 = b_3'$$

and  $a_{35} = +1$  or  $-1$ . Such a linear combination is given by row 3 minus row 1. Therefore,  $b_3' = b_3 - b_1$  and  $a_{35} = +1$ . In a similar fashion row 4 minus row 1 yields

$$-x_3 + x_6 = b_4 - b_1$$

The complete graph and  $N_d^-$  is given below.

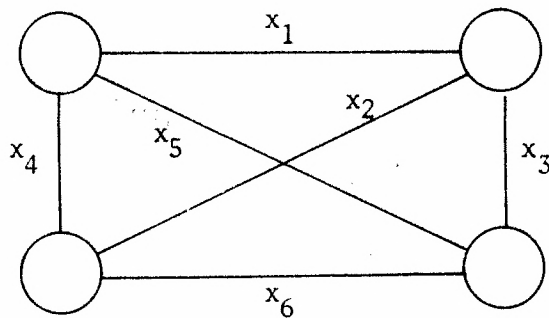


$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$\underline{b}$
$\left[ \begin{array}{c} 1 \\ -1 \\ 0 \\ 0 \end{array} \right]$	$\begin{array}{c} 1 \\ 0 \\ -1 \\ 0 \end{array}$	$\begin{array}{c} 1 \\ 0 \\ 0 \\ -1 \end{array}$	$\begin{array}{c} 0 \\ -1 \\ 0 \\ 0 \end{array}$	$\begin{array}{c} 0 \\ 0 \\ 1 \\ 0 \end{array}$	$\begin{array}{c} 0 \\ 0 \\ 0 \\ 1 \end{array}$	$\begin{array}{c} b_1 \\ -b_2 \\ b_3 - b_1 \\ b_4 - b_1 \end{array}$

Although the matroid corresponding to a matrix A of zeros and ones may be graphic, finding an equivalent network programming problem may not be possible. Consider the following.

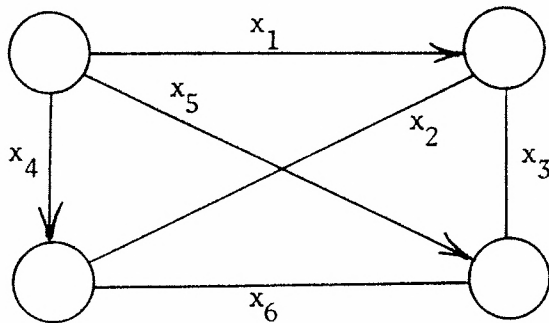
$$A = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{matrix}$$

A is graphic with the following graph and incidence matrix:



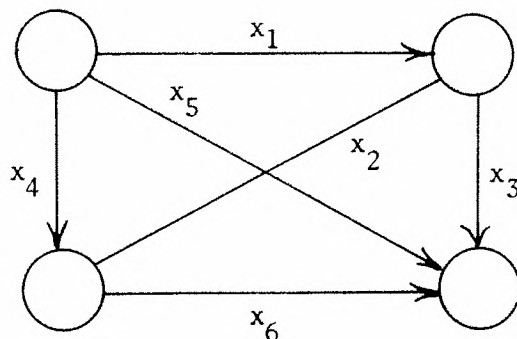
$$N = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

The first row of N is the same as the first row of A. Thus we have



The third row of N is the same as the third row of A. The coefficient

of  $x_5$  must be -1. We obtain, by multiplying this row by -1



The current matrix is

$$\begin{array}{cccccc}
 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\
 \begin{bmatrix}
 1 & 0 & 0 & 1 & 1 & 0 \\
 0 & 1 & 0 & 1 & 0 & 1 \\
 0 & 0 & -1 & 0 & -1 & -1 \\
 1 & 1 & 1 & 0 & 0 & 0
 \end{bmatrix}
 \end{array}$$

But now there exists no linear combination of rows of  $A$  with  $+1, 0$  coefficients that yield row 2 of  $N$  with the coefficient of  $x_4$  equal to -1. Nevertheless,  $A$  is totally unimodular! Hence it is not necessary that  $A$  can be transformed into the incidence matrix of a directed network in order that  $A$  be totally unimodular.

We now wish to show that if the matroid is not graphic, then such a transformation is impossible. Therefore it is necessary, but not sufficient, that the corresponding matroid be graphic.

### Theorem 2.12

A necessary condition that a matrix  $A$  of 0's and 1's be transformable





zero or an element of opposite sign than that of cell  $(i, \ell)$ . Otherwise we would have a submatrix of the form

$$\begin{array}{cc} & \begin{array}{cc} \ell & j \end{array} \\ \begin{array}{c} i \\ k \end{array} & \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \end{array}$$

Pivoting on cell  $(i, j)$  would result in a 2 in cell  $(k, \ell)$ , contradicting the hypothesis. Therefore if columns  $j$  and  $\ell$  contain non-zero elements in rows  $k$  and  $i$ , they will cancel upon pivoting. But this is precisely what happens if the signs are ignored and pivoting was done modulo 2. Non-zero elements will occur in the same positions. By replacing the -1's in  $N_d^-$  by 1's we have  $N^-$  of an undirected graph which would have been obtained by reducing  $A \bmod 2$  by the same sequence of operations.

#### A Unimodularity Theorem for the Multicommodity Transportation Problem

The MCTP was formulated in Chapter I. For an  $r$ -commodity,  $m$ -source,  $n$ -sink problem, denoted by  $\text{MCTP}(m, n, r)$ , the constraint matrix has the following form

$$A = \begin{bmatrix} A_1 & & & & \\ & A_2 & & & \\ & & \ddots & & \\ & & & A_r & \\ I & I & \dots & I & I \end{bmatrix}$$

Each  $A_k$  is an  $m$ -source,  $n$ -sink transportation problem constraint matrix. The identity matrices represent the capacity constraints.

The remainder of this chapter applies the previously surveyed results to produce a constructive proof of the following.

#### Theorem 2.13

A necessary and sufficient condition for the constraint matrix of an  $m$ -source,  $n$ -sink,  $r$ -commodity transportation problem to be totally unimodular is that either  $m \leq 2$  or  $n \leq 2$  when  $r \geq 2$ .

Reban [100] has investigated total unimodularity in the two-commodity transportation problem and established a somewhat weaker result than Theorem 2.13. His result is the following.

#### Theorem 2.14 (Reban [100])

Let  $A^*$  be the constraint matrix for a two-commodity transportation problem in which the capacitated edges form a tree with at most one interior node (i.e., all capacitated edges are incident with a common node). Then  $A^*$  is totally unimodular.

Note that Reban's theorem applies to the  $MCTP(m,n,2)$  so essentially is a different result. In our theorem, we are assuming that all  $u_{ij} < \infty$ . However, for the  $MCTP(2,n,2)$  or  $MCTP(m,2,2)$ , Theorem 2.13 is stronger since it allows all  $u_{ij} < \infty$  whereas Reban's result requires that only a subset of the arc capacities be finite.

To prove Theorem 2.13 we will first show that the  $MCTP(m,n,r)$  corresponds to a graphic matroid if and only if  $m \leq 2$  or  $n \leq 2$ , and then show how the structure of this matroid leads to the construction

of a single commodity flow problem equivalent to the MCTP( $m, n, r$ ) for  $m \leq 2$  or  $n \leq 2$ .

Theorem 2.15

The representative matrix of an MCTP( $m, n, r$ ) is the representative matrix of a graphic matroid if and only if  $m \leq 2$  or  $n \leq 2$ .

The proof is a direct application of Tutte's algorithm. The reader is advised to consult the Appendix before proceeding in order to familiarize himself with the terminology. Our aim is to show that the MCTP( $2, n, r$ ) and MCTP( $m, 2, r$ ) are representative matrices of graphic matroids. In general, the constraint matrix for MCTP( $m, n, r$ ) has the form

$$A = \begin{bmatrix} A_1 & & & & & \\ & A_2 & & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & A_r & \\ I & I & \dots & I & I \end{bmatrix}$$

where each  $A_k$  is the constraint matrix for MCTP( $m, n, 1$ ) and  $I$  is an  $(mn)$ th identity matrix. Since the rows of each  $A_k$  are linearly dependent, by convention let us drop the last row of  $A_k$  for all  $k$ . The remaining rows of  $A$  will be linearly independent. We will call this reduced matrix  $R$ , the representative matrix of MCTP( $m, n, r$ ).

The proof is by induction. We will first establish the result

for  $\text{MCTP}(2,2,2)$  and induce the result for  $m$  and  $r$ .

Lemma 2.1 The representative matrix of  $\text{MCTP}(2,2,2)$  is the representative matrix of a graphic matroid.

Proof. The representative matrix for  $\text{MCTP}(2,2,2)$  is  $R = R(2,2,2)$

$$R(2,2,2) = \begin{matrix} & \begin{matrix} 1 & 11 & 3 & 2 & 4 & 12 & 6 & 5 & 7 & 8 & 9 & 10 \end{matrix} \\ \begin{bmatrix} 1 & 1 & & & & & & & & & & \\ & & 1 & 1 & & & & & & & & \\ 1 & & 1 & & & & & & & & & \\ & & & 1 & 1 & & & & & & & \\ & & & & & 1 & 1 & & & & & \\ & & & & & & 1 & 1 & & & & \\ & & & & 1 & & 1 & & & & & \\ 1 & & & & 1 & & & & 1 & & & \\ & & 1 & & & 1 & & & & 1 & & \\ & & & 1 & & & 1 & & & & 1 & \\ & & & & 1 & & & 1 & & & & 1 \end{bmatrix} \end{matrix}$$

Rearranging the columns of  $R$  so that the first ten columns contain 1's along the diagonal yields

$$R = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{matrix} \\ \begin{bmatrix} 1 & & & & & & & & & & 1 & \\ & 1 & 1 & & & & & & & & & \\ 1 & & 1 & & & & & & & & & \\ & & & 1 & & & & & & & & 1 \\ & & & & 1 & 1 & & & & & & \\ & & & & 1 & & 1 & & & & & \\ 1 & & & 1 & & & & 1 & & & & \\ & & 1 & & & 1 & & & 1 & & 1 & 1 \\ & & & 1 & & & 1 & & & 1 & & \\ & 1 & & & 1 & & & & & & 1 & \end{bmatrix} \end{matrix}$$

We may now reduce  $R$  using modulo 2 arithmetic to its standard representative matrix  $R_1$ .

$$R_1 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} & \begin{bmatrix} 1 & & & & & & & & & & 1 & \\ & 1 & & & & & & & & & 1 & \\ & & 1 & & & & & & & & 1 & \\ & & & 1 & & & & & & & & 1 \\ & & & & 1 & & & & & & & 1 \\ & & & & & 1 & & & & & & 1 \\ & & & & & & 1 & & & & 1 & 1 \\ & & & & & & & 1 & & & 1 & 1 \\ & & & & & & & & 1 & & 1 & 1 \\ & & & & & & & & & 1 & 1 & 1 \end{bmatrix} \end{matrix}$$

Note that we may consider columns 1-10 as a basis of  $R$  modulo 2.

We are now in a position to apply Tutte's algorithm. Select any column with more than two 1's. Consider column 12. The first row with a one in this column is the fourth. Let this correspond to a point  $Y$  of  $M_1$ . (In general  $M_k$  is the matroid of the chain group defined by  $R_k$ ). Striking out row 4 and every column having a one in this row, we obtain the standard representative matrix  $R_2$  of  $M_1 \cdot (M_1 - Y)$ .



$$R_5 = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 7 \\ 8 \\ 9 \\ 10 \\ 4 \end{matrix} \begin{bmatrix} 1 & & & & & & & & & & 1 & \\ & 1 & & & & & & & & & 1 & \\ & & 1 & & & & & & & & 1 & \\ & & & 1 & & & & & & & 1 & 1 \\ & & & & & & 1 & & & & 1 & 1 \\ & & & & & & & 1 & & & 1 & 1 \\ & & & & & & & & 1 & & 1 & 1 \\ & & & & & & & & & 1 & 1 & 1 \\ & & & & & & & & & & & 1 \end{bmatrix} \end{array}$$

If zero columns are ignored, these are all standard representative matrices, to within a permutation of rows. Let the corresponding matroids be  $M_3$ ,  $M_4$ , and  $M_5$  respectively.

We now ask if  $B_1$ ,  $B_2$ , and  $B_3$  partition  $Y$ . Consider  $B_3$ . Strike out all columns of  $R_5$  having a zero in the last row, obtaining the following representative matrix of  $M_5 \cdot Y = (M_1 \chi(B_3 UY)) \cdot Y$ .

$$\begin{array}{c} \begin{matrix} 4 & 12 \end{matrix} \\ \begin{matrix} 7 \\ 8 \\ 9 \\ 10 \\ 4 \end{matrix} \begin{bmatrix} & 1 \\ & 1 \\ & 1 \\ & 1 \\ 1 & 1 \end{bmatrix} \end{array}$$

This may be reduced to standard form by adding the first row to all the remaining rows, thus obtaining



$$\begin{array}{cc}
 & 4 & 12 \\
 \begin{array}{c} 7 \\ 8 \\ 9 \\ 10 \\ 4 \end{array} & \left[ \begin{array}{cc} & 1 \\ & \\ & \\ & \\ 1 \end{array} \right]
 \end{array}$$

This standard representative matrix has only one 1 in each column. We may assert that  $B_3$  partitions  $Y$ . The corresponding partition is  $\{\{4\}, \{12\}\}$ . In a similar manner, we find that  $B_1$  and  $B_2$  determine the same partition. If the three bridges had not all partitioned  $Y$ , we would terminate and conclude that  $M_1$  is non-graphic. Since this is not the case, we must determine whether or not  $Y$  is even.  $B_1$  and  $B_2$  do not overlap since the union of the member  $\{4\}$  of  $B_1$  and  $\{12\}$  of  $B_2$  is all of  $Y$ . It follows that  $Y$  is even; its bridges can be arranged into two disjoint classes  $U = \{B_1, B_2\}$  and  $V = \{B_3\}$  so that no two members of the same class overlap.

We may now assert that  $M_1$  is graphic if and only if  $M_3$ ,  $M_4$ , and  $M_5$  are all graphic.  $M_3$  and  $M_4$  are graphic since  $R_3$  and  $R_4$  both have at most two 1's in each column. We now reapply the algorithm to  $R_5$ .

Consider column 12. Let row 7 correspond to a point  $Y$  of  $M_5$ . Striking out row 4 and all columns having a one in this row we obtain  $R_6$  of  $M_5 \cdot (M_5 - Y)$ .

$$\begin{array}{c}
 1 \quad 2 \quad 3 \quad 4 \quad 8 \quad 9 \quad 10 \\
 \begin{array}{c} 1 \\ 2 \\ 3 \\ 8 \\ 9 \\ 10 \\ 4 \end{array} \left[ \begin{array}{ccccccc} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ & & & & & & 1 \end{array} \right]
 \end{array}$$

The bridges of  $Y$  in  $M_5$  are

$$B_4 = \{1\}$$

$$B_5 = \{2\}$$

$$B_6 = \{3\}$$

$$B_7 = \{8\}$$

$$B_8 = \{9\}$$

$$B_9 = \{10\}$$

$$B_{10} = \{4\}$$

The  $Y$ -components corresponding to these bridges have the following matrices. ( $R_7$  is associated with  $B_4$ ,  $R_8$  with  $B_5$ , etc.)

$$\begin{array}{c}
 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \quad 11 \quad 12 \\
 R_7 = \frac{1}{7} \left[ \begin{array}{cccccccccccc} 1 & & & & & & & & & & 1 & \\ & & & & & & & & & & 1 & 1 \end{array} \right] \\
 R_8 = \frac{2}{7} \left[ \begin{array}{cccccccccccc} & 1 & & & & & & & & & 1 & \\ & & & & & & & & & & 1 & 1 \end{array} \right]
 \end{array}$$

$$\begin{array}{c}
 \begin{array}{cccccccccc}
 & 1 & 2 & 3 & 4 & 8 & 9 & 10 & & \\
 1 & & & & & & & & & \\
 2 & & & & & & & & & \\
 3 & & & & & & & & & \\
 8 & & & & & & & & & \\
 9 & & & & & & & & & \\
 10 & & & & & & & & & \\
 4 & & & & & & & & & 
 \end{array}
 \begin{bmatrix}
 1 & & & & & & & & & \\
 & 1 & & & & & & & & \\
 & & 1 & & & & & & & \\
 & & & 1 & & & & & & \\
 & & & & 1 & & & & & \\
 & & & & & 1 & & & & \\
 & & & & & & 1 & & & \\
 & & & & & & & 1 & & \\
 & & & & & & & & 1 & \\
 & & & & & & & & & 1
 \end{bmatrix}
 \end{array}$$

The bridges of  $Y$  in  $M_5$  are

$$B_4 = \{1\}$$

$$B_5 = \{2\}$$

$$B_6 = \{3\}$$

$$B_7 = \{8\}$$

$$B_8 = \{9\}$$

$$B_9 = \{10\}$$

$$B_{10} = \{4\}$$

The  $Y$ -components corresponding to these bridges have the following matrices. ( $R_7$  is associated with  $B_4$ ,  $R_8$  with  $B_5$ , etc.)

$$\begin{array}{c}
 \begin{array}{cccccccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & \\
 R_7 = \frac{1}{7} \begin{bmatrix}
 1 & & & & & & & & & & & & & \\
 & & & & & & & & & & & 1 & & \\
 & & & & & & & & & & & 1 & & \\
 & & & & & & & 1 & & & & 1 & 1 & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & 
 \end{bmatrix} \\
 R_8 = \frac{2}{7} \begin{bmatrix}
 & & & & & & & & & & & 1 & & \\
 & & & & & & & & & & & 1 & & \\
 & & & & & & & & & & & 1 & & \\
 & & & & & & & 1 & & & & 1 & 1 & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & 
 \end{bmatrix}
 \end{array}
 \end{array}$$

$$\begin{array}{c}
 \begin{array}{cccccccccccccc}
 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12
 \end{array} \\
 R_9 = \begin{array}{c} 3 \\ 7 \end{array} \begin{bmatrix} & & 1 & & & & & & & & 1 & \\ & & & & & & 1 & & & & 1 & 1 \end{bmatrix} \\
 R_{10} = \begin{array}{c} 8 \\ 7 \end{array} \begin{bmatrix} & & & & & & & 1 & & & 1 & 1 \\ & & & & & & 1 & & & & 1 & 1 \end{bmatrix} \\
 R_{11} = \begin{array}{c} 9 \\ 7 \end{array} \begin{bmatrix} & & & & & & & & 1 & & 1 & 1 \\ & & & & & & 1 & & & & 1 & 1 \end{bmatrix} \\
 R_{12} = \begin{array}{c} 10 \\ 7 \end{array} \begin{bmatrix} & & & & & & & & & 1 & 1 & 1 \\ & & & & & & 1 & & & & 1 & 1 \end{bmatrix} \\
 R_{13} = \begin{array}{c} 4 \\ 7 \end{array} \begin{bmatrix} & & & 1 & & & & & & & & 1 \\ & & & & & & 1 & & & & 1 & 1 \end{bmatrix}
 \end{array}$$

These correspond to  $M_7$  through  $M_{13}$  respectively. It should be clear that  $B_4$  through  $B_{10}$  partition  $Y$ . The partitions  $P_i$  corresponding to  $B_i$  are

$$P_4 = \{\{7,12\}, \{11\}\}$$

$$P_5 = \{\{7,12\}, \{11\}\}$$

$$P_6 = \{\{7,12\}, \{11\}\}$$

$$P_7 = \{\{7\}, \{11,12\}\}$$

$$P_8 = \{\{7\}, \{11,12\}\}$$

$$P_9 = \{\{7\}, \{11,12\}\}$$

$$P_{10} = \{\{7,11\}, \{12\}\}$$



$$R_1(k,2,2) = \begin{array}{c|cccccc} & p & q & r & s & u & v \\ \hline & & & & & 1 & \\ & & & & & 1 & \text{kth row} \\ & & & & & 1 & 1 \\ & & & & & 1 & 1 \text{ 2kth row} \\ & & & & & 1 & \\ & & & & & 1 & \\ \hline p & & & & & 1 & \\ q & & & & & & 1 \\ r & & & & & 1 & 1 \\ s & & & & & 1 & 1 \end{array}$$

← Y

To verify the form of this matrix, note that

$$R(k-1,2,2) = \begin{bmatrix} P & & \\ & P & \\ I & I & I \end{bmatrix}$$

where P is of the form

$$\left. \begin{array}{l} k-1 \text{ rows} \end{array} \right\} \begin{bmatrix} \textcircled{1} & 1 & & & \\ & \textcircled{1} & 1 & & \\ & & & \ddots & \\ & & & & \ddots & \\ & & & & & 1 & \textcircled{1} \\ 1 & 1 & \dots & \textcircled{1} \end{bmatrix}$$

The circled elements form the basis by our construction. As  $R(k-1,2,2)$  is reduced to standard form, elimination of the other ones in the basic columns leaves the following matrix in the last six columns with the last four rows also reduced.

	p	q	r	s	u	v	
	1						kth row
	1						
	.						
	.						
		1					2kth row
		1					
		.					
		.					
	1	1					4kth row
	1	1					
	1	0	0	0	1	0	
	0	1	0	0	1	1	
	0	0	1	0	1	1	
	0	0	0	1	0	1	

The ones in rows  $4k$  and  $4k-1$  arise from eliminating the ones in the identity matrix using the basic elements in the last two rows of  $P$ . Now elimination of the ones in the first  $4k$  rows of columns  $p$  and  $q$  leaves the resulting matrix  $R_1(k,2,2)$ . Letting row  $q$  be a point  $Y$  of  $M$ , we find that the bridges of  $M \cdot (M - Y)$  are simply

$$B_1 = \{p\}$$

$$B_2 = \{r\}$$

$$B_3 = \{s\}$$

$$B_4 = \text{bridges obtained from } R_1(k-1,2,2)$$

We may form the Y-components corresponding to these bridges by adjoining  $q$  to the rows of  $R_1(k,2,2)$  that determine these bridges. From the proof of Lemma 2.1, it should be obvious that  $B_1$ ,  $B_2$ , and  $B_3$  partition  $Y$ , determining partitions

$$P_1 = \{\{q,v\}, \{u\}\}$$

$$P_2 = \{\{q\}, \{u,v\}\}$$

$$P_3 = \{\{q,u\}, \{v\}\}$$

To determine  $P_4$  we must characterize  $B_4$ . We do this with the following lemma.

Lemma 2.3 The columns associated with  $R_1(k-1,2,2)$  form  $B_4$ .

Proof. The general rule for constructing an elementary separator is described in [128]. Take an arbitrary row of  $R_1(k,2,2)$ , then every row having a 1 in one of the same columns as the first row taken, then every row having a 1 in the same column as a row already chosen, and so on. The elementary separator is determined by the 1's of the resulting submatrix. Refer to  $R_1(2,2,2)$  in the proof of Lemma 2.1. Consider  $R_1(3,2,2)$ . Clearly the columns associated with  $R_1(2,2,2)$  determine a bridge of  $R_1(3,2,2)$  as can be seen from the construction of  $R_1(k,2,2)$ . Now assume the lemma true that  $R_1(k-2,2,2)$  is a bridge of  $R_1(k-1,2,2)$ . But observe that in  $R_1(k-1,2,2)$  that the  $2k$ th row has a 1 in columns  $u$  and  $v$ , and that columns  $u$  and  $v$  have a 1 in rows  $p$ ,  $q$ ,  $r$ , and  $s$ . Hence, using the rule described above, all columns



will appear in the description of the elementary separator. It follows that the columns of  $R_1(k-1,2,2)$  form a bridge of  $R_1(k,2,2)$ .

Returning to the proof of Lemma 2.2, we have that  $B_4$  consists of all columns of  $R_1(k-1,2,2)$ .  $B_4$  partitions  $Y$  and the corresponding partition is

$$P_4 = \{\{u\}, \{v\}, \{q\}\}$$

Letting  $U = \{B_1, B_2, B_3\}$  and  $V = \{B_4\}$  we see that  $Y$  is even. Since  $B_1$ ,  $B_2$ , and  $B_3$  consist of a single element, it readily follows that their corresponding matroids are graphic because each  $Y$ -component has at most two 1's in every column. Since the matroid for the  $(k-1,2,2)$  problem is graphic by hypothesis, we conclude that the matroid for the  $(k,2,2)$  problem is graphic, Q.E.D.

Lemma 2.4 The representative matrix for  $MCTP(m,2,r)$ ,  $m, r \geq 2$ , is the representative matrix of a graphic matroid.

Proof. From Lemma 2.2, the result is true for  $r = 2$ . Assume it is true for  $r = k-1$ . We prove it true for  $r = k$ . The representative matrix  $R(m,2,k)$  has the form

$$R(m,2,k) = \begin{bmatrix} A_1^* & & & & \\ & A_2^* & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & A_k^* \\ I & I & \dots & I & I \end{bmatrix}$$

where  $A_j^*$  is  $A$  with the last row deleted, and  $I$  is a unit matrix of dimension  $2m$ . We may permute the rows and columns of  $R(m,2,k)$  to put it into the following form.

$$R(m,2,k) = \begin{array}{|c|c|} \hline \begin{array}{|c|} \hline \begin{array}{c} A_1^* \\ \\ A_2^* \\ \\ \cdot \\ \cdot \\ \cdot \\ \\ A_{k-1}^* \end{array} \\ \hline \end{array} & \begin{array}{|c|} \hline \\ \hline \end{array} \\ \hline \begin{array}{|c|} \hline I \quad I \quad \dots \quad I \quad I \\ \hline \end{array} & \begin{array}{|c|} \hline I \\ \hline \end{array} \\ \hline \begin{array}{|c|} \hline \underline{0} \\ \hline \end{array} & \begin{array}{|c|} \hline A_k \\ \hline \end{array} \\ \hline \end{array}$$

The matrix delineated by double lines is the representative matrix  $R(m,2,k-1)$ . Suppose we reduce  $R(m,2,k)$  to its standard representative matrix. Compute the standard representative matrix for  $R(m,2,k-1)$  in the usual manner. This leaves the last  $2m$  columns of  $R(m,2,k)$  unaffected. To form the complete standard representative matrix for  $R(m,2,k)$ , we need a basis for  $R(m,2,k-1)$  and  $A_k$ . Since  $A_1, A_2, \dots, A_k$  are identical, the process of row reduction mod 2 will yield the following standard representative matrix  $R_1(m,2,k)$ :

$$R_1(m, 2, k) =$$

$I_{m+1}$ $I_{m+1}$ $I_{m+1}$ $I_{m+1}$ $I_{2m}$	$N$ $N$ $N$ $N$		
	$Q$	$Q$	$Q$
			$Q$
		$I_{m+1}$	$N$

Now consider choosing a point  $Y$  from among the last  $m+1$  rows. We obtain the elementary separators of  $M \cdot (M - Y)$ . We ask if these bridges partition  $Y$ . From the  $Y$ -components formed from each bridge, we see that upon striking out all columns having a zero in the row corresponding to  $Y$ , we are left with precisely the same matrix as if we had chosen  $Y$  from some similar set of rows corresponding to  $A_1^*, A_2^*, \dots, A_{k-1}^*$ , since each set of rows contains  $I_{m+1}$  and  $N$ , up to a permutation of rows and columns. But by hypothesis, the matroid of the  $MCTP(m, 2, k-1)$  is graphic, and the  $Y$  we chose from among the rows corresponding to  $A_k^*$  determines the same form of the partitions of its bridges as if we would have chosen  $Y$  among the rows corresponding to  $A_j^*$ ,  $j < k$ . All that differs is the names of the columns. Since indices are not relevant, it is clear that  $Y$  is even and hence the matroid is graphic.

To complete the sufficiency proof of Theorem 2.15, observe that the representative matrix of an  $MCTP(m, 2, k)$  is identical to that of an  $MCTP(2, m, k)$  to within a permutation of rows and columns. This is easily seen by simply renaming the sources and sinks of the multi-commodity transportation problem.



Let  $Y$  correspond to row 17. Striking out this row and columns 21, 24, 25, 26, and 27, we obtain  $R_2$  of  $M \cdot (M - Y)$ .

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	22	23
$R_2 =$	1														1							
		1														1						
			1													1	1					
				1														1				
					1													1	1			
						1												1	1			
							1								1		1					
								1							1		1					
									1							1		1				
										1							1		1			
											1							1				
												1							1			
													1							1		
														1							1	
															1	1	1	1				
															1	1	1	1				
																			1			
																				1		
																					1	
																						1

The bridges of  $Y$  in  $M$  are

$$B_1 = \{19\}$$

$$B_2 = \{20\}$$

$$B_3 = \{22\}$$

$$B_4 = \{23\}$$

$$B_5 = \{1, 2, \dots, 18\}$$

We find that the partitions determined by these bridges are

$$P_1 = \{\{21,26,27\}, \{24,25\}\}$$

$$P_2 = \{\{21,24,25\}, \{26,27\}\}$$

$$P_3 = \{\{24,26\}, \{21,25,27\}\}$$

$$P_4 = \{\{25,27\}, \{21,24,26\}\}$$

$$P_5 = \{\{24\}, \{25\}, \{26\}, \{27\}, \{21\}\}$$

Observe that it is impossible to arrange  $B_1, \dots, B_5$  into two disjoint classes such that no two members of the same class overlap. Hence,  $Y$  is not even and  $M$  is therefore not graphic. Since the representative matrix of an MCTP(3,3,2) is a submatrix for any larger problem, necessity follows.

### Matroid Graphs of Graphic MCTP Matroids

We now consider the question of constructing the associated matroid graphs. Tutte gives some general guidelines in [128] that rely on his algorithm; but much of the procedure is trial and error.

Since each row of a standard representative matrix represents a bond in a graphic matroid, one method of constructing a graph is to draw arcs in such a way that each cut set is preserved in the graph. Since the rows of a standard representative matrix are linearly independent, the number of nodes is equal to the number of rows plus one. Once an initial graph is drawn, one may verify the construction by checking the circuits in the last  $n - m$  columns of the standard representative matrix. If all circuits are represented, then the graph is a true representation of the matrix.

Consider  $R_1(2,2,2)$  in the proof of the previous section. The

elementary cycles of  $G$  are

$$C_1 = \{1, 2, 3, 7, 8, 9, 10, 11\}$$

$$C_2 = \{4, 5, 6, 7, 8, 9, 10, 12\}$$

The graph of  $R_1(2,2,2)$  is given in Figure 12. Each row of  $R_1(2,2,2)$  can be verified to be a cut set of this graph. It is interesting to note that  $G(2,2,2)$  is planar; hence its dual graph  $G^*(2,2,2)$  exists. From a theorem of Tutte [131], the matroid of  $R_1(2,2,2)$  is also co-graphic.  $G^*(2,2,2)$  is given in Figure 13.

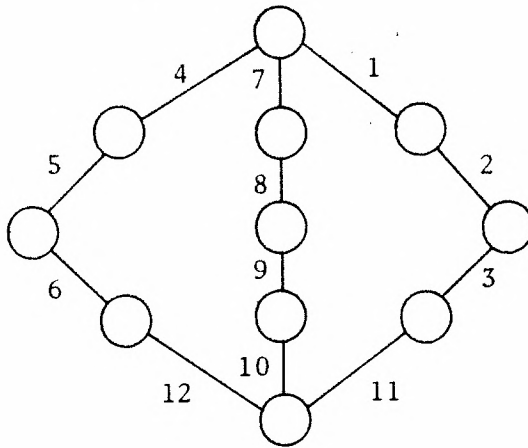


Figure 12. The Graph  $G(2,2,2)$

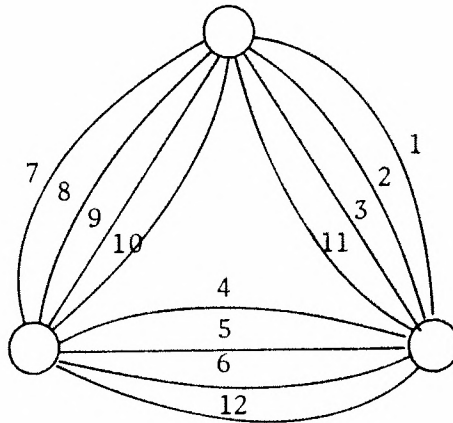


Figure 13. The Graph  $G^*(2,2,2)$

We may characterize those graphs for  $(2,2,r)$  problems. Recall that every cut set of  $G$  is a cycle of  $G^*$  and conversely. Now consider  $R(2,2,r)$ .

$$R(2,2,r) = \begin{bmatrix} A_1^* & & & & & \\ & A_2^* & & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & A_r^* & \\ I_4 & I_4 & \dots & I_4 & I_4 \end{bmatrix}$$

where each  $A_i = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$

and  $I_4$  is an order 4 identity matrix. The standard representative matrix of  $R_1(2,2,r)$  has the form

$$\left[ \begin{array}{cccc|c} I_3 & & & & \underline{1} \\ & I_3 & & & \underline{1} \\ & & \ddots & & \\ & & & I_3 & \\ & & & & \underline{1} \\ & & & I_4 & [1] \end{array} \right]$$

where  $\underline{1}$  is a  $3 \times 1$  vector of 1's and  $[1]$  denotes a matrix of all 1's. We will characterize  $G^*$ .  $G^*$  consists of  $r+1$  vertices and has the basic form of a regular polygon. Each set of three rows of  $R_1(2,2,r)$  typically looks like



$$\begin{array}{c}
 \begin{array}{ccccc}
 & q & r & s & & p \\
 q & \left[ \begin{array}{c|c|c|c|c}
 & 1 & & & 1 \\
 \hline
 r & \underline{0} & & 1 & \underline{0} \\
 \hline
 s & & & 1 & 1 \\
 \hline
 & & & & 1 \\
 \hline
 & & & & \underline{0}
 \end{array} \right]
 \end{array}
 \end{array}$$

and defines three cycles of  $G^*$ . The set of arcs joining two vertices of  $G^*$  is  $\{p, q, r, s\}$ . Denote each of these sets by  $L_k$ ,  $k = 1, 2, \dots, r$ . Denote the set of row indices of the last four rows of  $R_1(2, 2, r)$  by  $L$ . Then  $G^*$  has the form of the graph in Figure 14 where  $L$  and  $L_k$  denote sets of arcs.

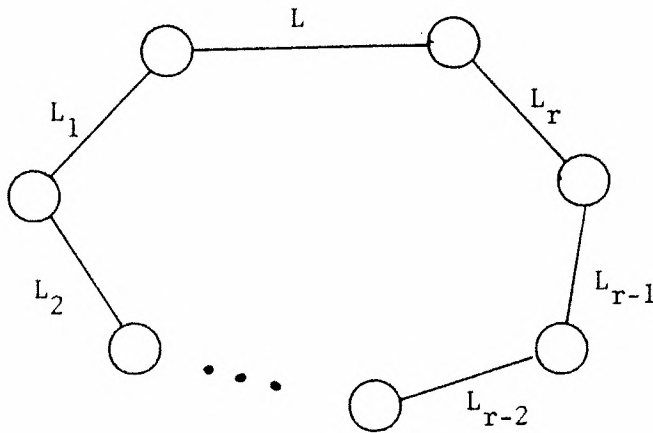


Figure 14. The Graph  $G^*(2, 2, r)$

In terms of the original graph  $G(2, 2, r)$  this is equivalent to adding a set of four edges for each commodity to the graph in Figure 12. For example,  $G(2, 2, 3)$  is shown in Figure 15.

The matroid graph of any other larger problem is non-planar, as seen by  $G(3, 2, 2)$  in Figure 16. The standard representative matrix  $R_1(3, 2, 2)$  is given below the figure and  $G(3, 2, 2)$  may be verified from it.

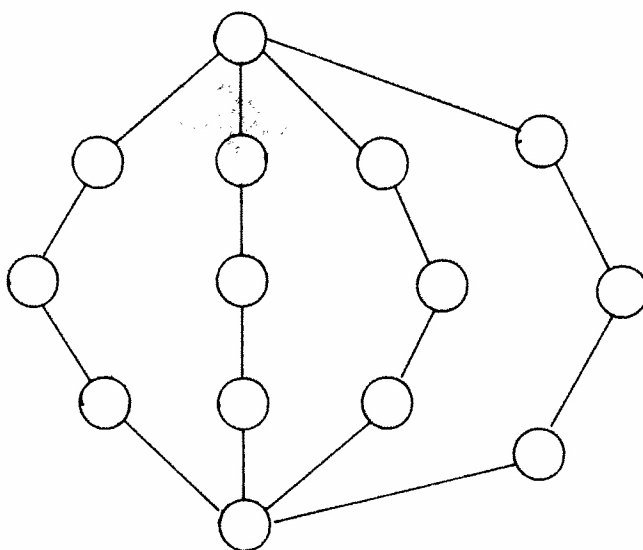


Figure 15. The Graph  $G(2,2,3)$

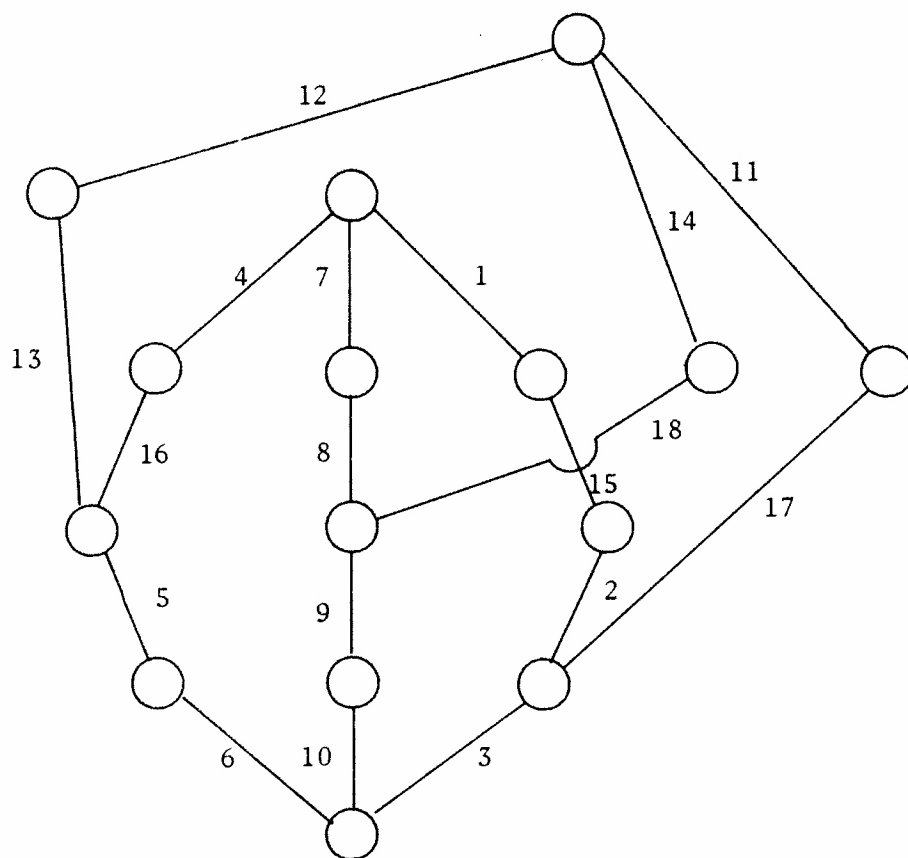


Figure 16. The Graph  $G(3,2,2)$

$$R_1(3,2,2) =$$

	1	2	3	4	5	6	7	8	9	10	15	16	11	12	13	14	17	18
1	1										1							
2		1									1						1	
3			1								1						1	
4				1								1						
5					1							1					1	1
6						1						1					1	1
7							1				1	1						
8								1			1	1						
9									1		1	1						1
10										1	1	1						1
11													1				1	
12														1			1	1
13															1		1	1
14																1		1

Recall that the representative matrix of an  $MCTP(m,2,r)$  problem is equivalent up to permutations of rows and columns to an  $MCTP(2,m,r)$  problem. Hence their corresponding matroid graphs are isomorphic. Notice that all the graphs constructed in this section are bipartite. In the next section we will show how to construct an equivalent single commodity directed network for the general  $MCTP(m,n,r)$ . This was developed from analysis of the structures of the graphs presented in this section, but the author could not rigorously prove that these are actually "matroid graphs."

#### An Equivalent Single Commodity Network for the $MCTP(m,2,r)$

The matroid graphs discussed in the previous section have a natural interpretation in terms of network linear programming. Consider

$G(2,2,2)$  and label the edges with the variables the arcs represent.

By multiplying certain rows of the node-arc incidence matrix of  $G(2,2,2)$  by  $-1$  we obtain the incidence matrix of a directed graph  $G'(2,2,2)$  in Figure 17.

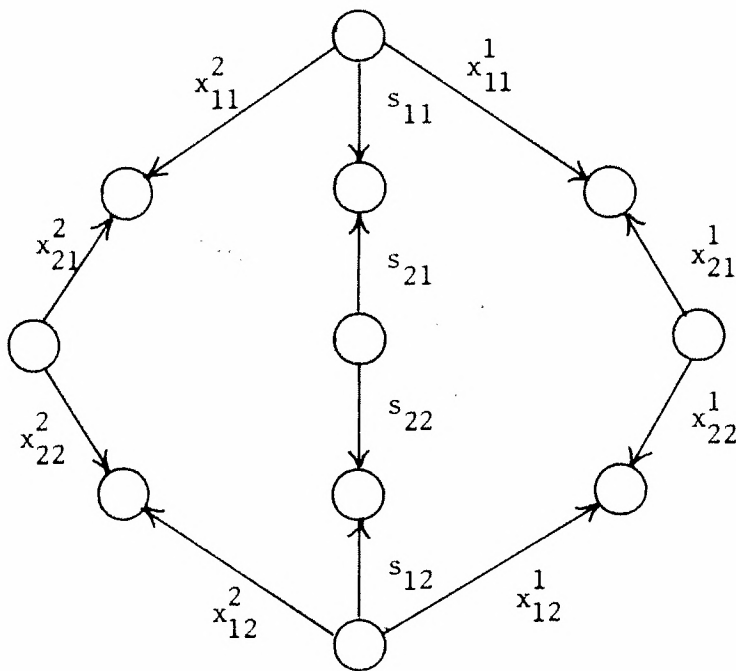


Figure 17. The Graph  $G'(2,2,2)$

In terms of the multicommodity transportation problem, the node-arc incidence matrix of  $G'(2,2,2)$  is the constraint matrix of the following linear program.

$$\text{minimize} \quad \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 c_{ij}^k x_{ij}^k$$

subject to

$$\begin{array}{rclcl}
 x_{11}^1 & & +x_{11}^2 & & +s_{11} & = & u_{11} \\
 -x_{11}^1 & & -x_{21}^1 & & & = & -b_1^1 \\
 & & x_{21}^1 & & +x_{22}^1 & = & a_2^1 \\
 -x_{12}^1 & & -x_{22}^1 & & & = & -b_2^1 \\
 x_{12}^1 & & +x_{12}^2 & & +s_{12} & = & u_{12} \\
 & & -x_{12}^2 & & -x_{22}^2 & = & -b_2^2 \\
 & & & & x_{21}^2 & +x_{22}^2 & = & a_2^2 \\
 & & -x_{11}^2 & & -x_{21}^2 & = & -b_1^2 \\
 & & & & -s_{11} & -s_{21} & = & b_1^1 + b_1^2 - u_{11} - u_{21} \\
 & & & & & s_{21} & +s_{22} & = & u_{21} + u_{22} - a_2^1 - a_2^2 \\
 & & & & -s_{12} & -s_{22} & = & b_2^1 + b_2^2 - u_{12} - u_{22}
 \end{array}$$

$x_{ij}^k, s_{ij} \geq 0$

This could also be obtained by multiplying the original MCTP(2,2,2) constraint matrix by an appropriate non-singular unimodular transformation matrix. Observe that the node conservation equations represent supplies, demands, capacities, or some linear combination of these. All constraints of the original problem are satisfied.

We will now describe an algorithm for constructing a single commodity transportation-type network for the general MCTP(2,m,r) problem. The algorithm is symmetric for the MCTP(2,m,r).

Algorithm 1

1. Create  $m$  source nodes for supplies  $a_i^k$ ,  $i = 1, m$ ;  $k = 1, r$ .
2. Create  $r$  sink nodes for demands  $b_1^k$ ,  $k = 1, r$ .
3. Create  $m$  sink nodes for capacities  $u_{i2}$ ,  $i = 1, m$ .
4. Create  $m$  source nodes, each having supply

$$\sum_{j=1}^2 u_{ij} - \sum_{k=1}^r a_i^k \quad \text{for } i = 1, m$$

5. Create a (redundant) sink node with demand

$$- \sum_{k=1}^r b_1^k + \sum_{i=1}^m u_{i1}$$

6. For each variable  $x_{ij}^k$  join an arc from source nodes created in step 1 to the appropriate sink node created in steps 2 or 3.
7. For each slack variable  $s_{ij}$ , join an arc from the proper node created in step 4 to the proper node created in step 3 or 5.

An example for the MCTP(3,2,2) is given in Figure 18.

Theorem 2.16

Algorithm 1 yields a network problem equivalent to the MCTP( $m, 2, r$ ).

Proof. By construction, the source nodes created in step 1 represent

$$\sum_{j=1}^2 x_{ij}^k = a_i^k \quad \text{for all } i, k \quad (2.1)$$

Steps 2, 3, 4, and 5 yield

$$\sum_{i=1}^m x_{i1}^k = b_1^k \quad \text{for all } k \quad (2.2)$$

$$\sum_{k=1}^r x_{i2}^k + s_{i2} = u_{i2} \quad \text{for all } i \quad (2.3)$$

$$\sum_{j=1}^2 u_{ij} - \sum_{k=1}^r a_i^k = \sum_{j=1}^2 s_{ij} \quad \text{for all } i \quad (2.4)$$

$$\sum_{i=1}^m s_{i1} = \sum_{k=1}^r b_1^k - \sum_{i=1}^m u_{i1} \quad (2.5)$$

The original MCTP constraints are

$$\sum_{j=1}^n x_{ij}^k = a_i^k \quad \text{for all } i, k \quad (2.6)$$

$$\sum_{i=1}^m x_{ij}^k = b_j^k \quad \text{for all } j, k \quad (2.7)$$

$$\sum_{k=1}^r x_{ij}^k + s_{ij} = u_{ij} \quad \text{for all } i, j \quad (2.8)$$

Equation (2.1) is precisely (2.6). System (2.6-2.8) is redundant by nature of the single commodity transportation problem constraints.

Let the redundant equations be

$$\sum_{i=1}^m x_{ij}^k = b_2^k \quad \text{for all } k \quad (2.9)$$

Then (2.2) is precisely (2.7) without the redundancy; (2.3) implies (2.8) for  $j = 2$ . Substitution of (2.1) into (2.4) and adding (2.3) yields (2.8) with  $j = 1$ . Equation (2.5) is linearly dependent on (2.1-2.4). This completes the proof.

By attaching costs of  $c_{ij}^k$  to arcs corresponding to  $x_{ij}^k$  and 0 to those corresponding to  $s_{ij}$ , we have a single commodity network problem that can be easily solved.

### Proof of Theorem 2.13

We are now in a position to prove the main result. Sufficiency is proven as follows: If  $A\underline{x} = \underline{b}$  is the original MCTP constraint set and  $A^*\underline{x} = \underline{b}^*$  is the constraint set for the equivalent network problem defined by Algorithm 1, we note that the components of  $\underline{b}^*$  are integer linear combinations of the components of  $\underline{b}$ . For all integer  $\underline{b}$ , we note that  $\{\underline{x} | A\underline{x} = \underline{b}, \underline{x} \geq 0\}$  has the integral property since the equivalent system  $\{\underline{x} | A^*\underline{x} = \underline{b}^*, \underline{x} \geq 0\}$  does. By Theorem 2.2, it follows that  $A$  is totally unimodular.

To prove necessity, it is easy to find a basis for the  $MCTP(m,n,r)$  with both  $m$  and  $n$  greater than 2 that has a determinant with absolute value greater than unity. We shall give an  $MCTP(3,3,2)$  example. Since the constraint matrix for an  $MCTP(3,3,2)$  is a submatrix for any larger problem, necessity readily follows. The  $MCTP(3,3,2)$  constraint matrix is



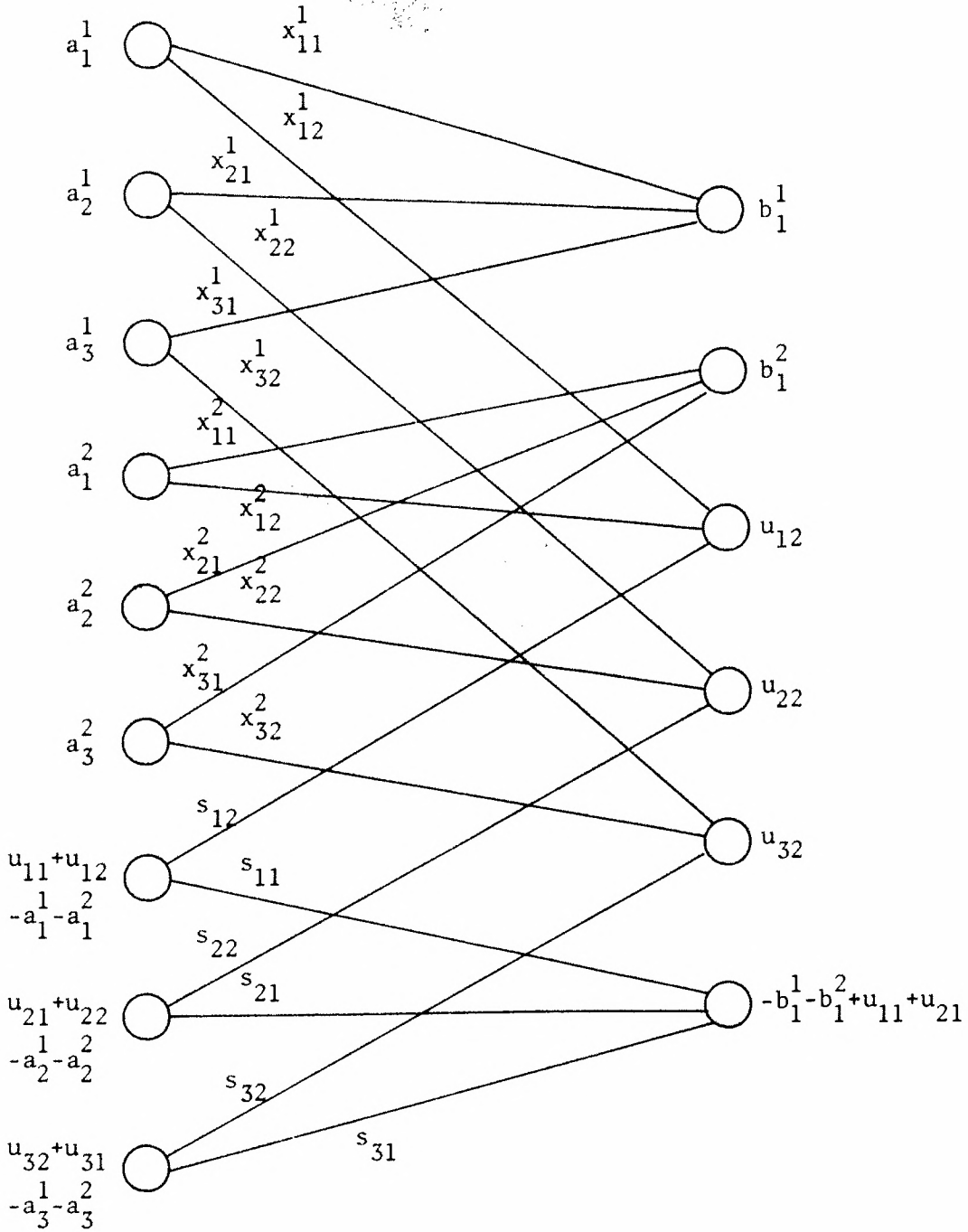
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1	1	1																								
			1	1	1																					
						1	1	1																		
1			1			1																				
	1			1			1																			
									1	1	1															
												1	1	1												
															1	1	1									
									1			1			1											
										1			1			1										
1									1										1							
	1									1										1						
		1									1										1					
			1									1										1				
				1									1										1			
					1									1										1		
						1									1										1	
							1									1										1
								1									1									
									1									1								
										1									1							
											1									1						
												1									1					
													1									1				
														1									1			
															1									1		
																1									1	
																	1									1

Choosing columns 1,3,5,6,7,9,10,11,13,15,17,18,20,21,22,23,24,25, and 26 as a basis  $B$ , it may be verified that  $\det B = 2$ .

We now wish to present another construction which solves the MCTP( $m,2,r$ ), and is smaller in size. This will be called a reduced matroid graph.

#### Algorithm 2

1. Create  $mr$  source nodes for supplies  $a_i^k$ , all  $i,k$ .
2. Create  $r$  sink nodes for demands  $b_1^k$ , all  $k$ .
3. Create  $m$  sink nodes for capacities  $u_{i2}$ , all  $i$ .

Figure 18. The Graph  $G'(3,2,2)$

4. Create a source node representing

$$\sum_i u_{i2} - \sum_k b_2^k$$

5. For each variable  $x_{ij}^k$  join an arc from source node  $a_i^k$  to sink node  $b_1^k$  (from step 2) or  $u_{i2}$  (step 3).
6. For slack variables  $s_{i2}$ , join an arc from the node created in step 4 to the appropriate node created in step 3.

An example for the MCTP(3,2,2) is illustrated in Figure 19. Note that the reduced matroid graphs correspond to a relaxation of the constraints

$$\sum_k x_{i1}^k \leq u_{i1} \quad \text{for all } i$$

since the slack variables  $s_{i1}$  are not explicitly represented.

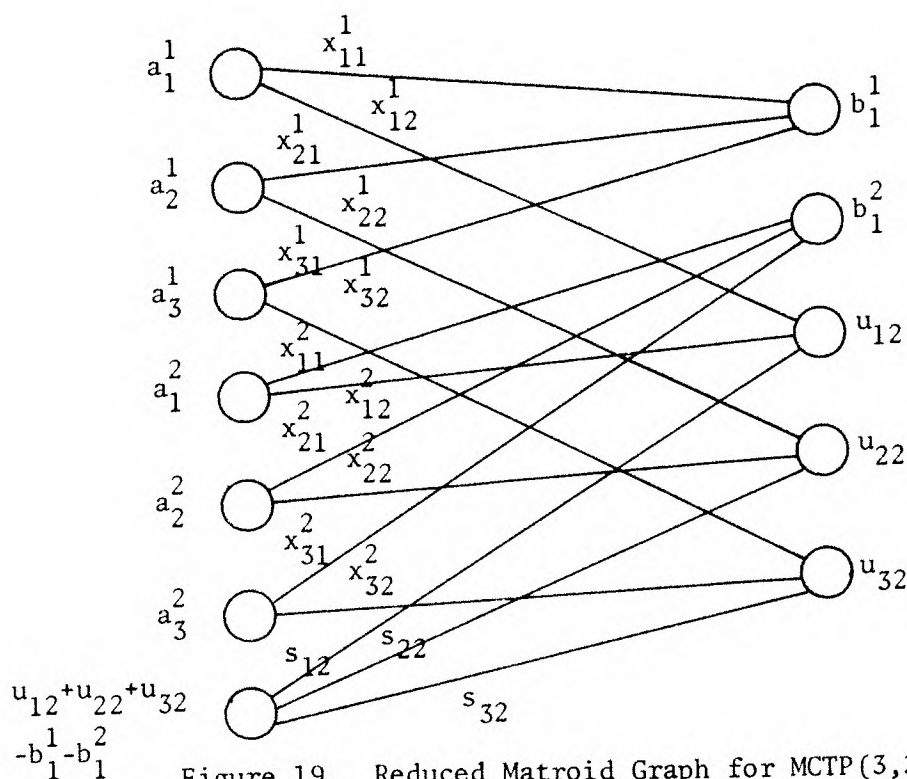


Figure 19. Reduced Matroid Graph for MCTP(3,2,2)

Theorem 2.17

Algorithm 2 yields a network problem equivalent to the MCTP(m,2,r).

Proof. We shall show that the constraints

$$\sum_k x_{i1}^k \leq u_{i1} \quad \text{for all } i$$

are satisfied in the reduced matroid graph. Assuming that a feasible solution exists, then

$$\sum_k x_{ij}^k + s_{ij} = u_{ij} \quad \text{for all } i, j$$

has the property that  $x_{ij}^k, s_{ij} \geq 0$ . Summing over  $j$  we have

$$\sum_j \sum_k x_{ij}^k + \sum_j s_{ij} = \sum_j u_{ij}$$

Clearly

$$\sum_k \sum_j x_{ij}^k + s_{i2} \leq \sum_j u_{ij}$$

Since

$$a_i^k = \sum_j x_{ij}^k \quad \text{we have}$$

$$\sum_k a_i^k + s_{i2} \leq u_{i1} + u_{i2}$$

$$\Rightarrow \sum_k a_i^k - (u_{i2} - s_{i2}) \leq u_{i1}$$

$$\Rightarrow \sum_k a_i^k - \sum_k x_{i2}^k \leq u_{i1}$$

$$\Rightarrow \sum_k (a_i^k - x_{i2}^k) \leq u_{i1}$$

$$\Rightarrow \sum_k x_{il}^k \leq u_{il} \quad \text{Q.E.D.}$$

In Chapter V we shall define a generalization of reduced matroid graphs for the MCTP(m,n,r) and see that this result is a special case of a more general theorem.

We wish to remark that in solving the MCTP(2,n,r), that reduced matroid graphs provide a slight computational advantage over standard multicommodity flow techniques in that less information must be stored and maintained. In particular, in the reduced matroid graph, there are  $mnr + m$  variables whereas a standard procedure must maintain  $mnr + 2m$  variables. The inverse is computed entirely by graph-theoretic means, whereas in a procedure such as Hartman and Lasdon [45], a "working basis" of dimension equal to the number of saturated arcs must be maintained explicitly. For these reasons, it would appear that reduced matroid graphs would be computationally more efficient.

## CHAPTER III

### A GRAPH-THEORETIC CHARACTERIZATION OF INTEGER BASIC FEASIBLE SOLUTIONS IN MULTICOMMODITY NETWORKS

#### Introduction

In this chapter we will discuss the basis characterization of multicommodity network flow problems. This will be used to interpret the integrality question in multicommodity networks from a graph-theoretic point of view, and establish a sufficient condition under which a multicommodity flow basis will be totally unimodular.

#### Single Commodity Bases

The graph-theoretic structure of bases in single commodity network flow problems has been known for a long time (see, for instance, Dantzig [17], Chapter 17), but Ellis Johnson first presented a unified treatment and computational procedure [69, 70]. We shall assume that the reader is familiar with the basic terminology of graph theory.

The network linear programming problem on a directed graph  $G$  is

$$\begin{array}{ll} \text{Minimize} & z = \underline{c} \underline{x} + \underline{0} \underline{s} \\ \text{subject to} & A\underline{x} + U\underline{s} = \underline{b} \\ & \underline{0} \leq \underline{x} \leq \underline{\alpha} \\ & \underline{0} \leq \underline{s} \leq \underline{\sigma} \end{array}$$

where  $A$  is the  $m \times n$  node-arc incidence matrix of  $G$ , and  $U$  is an

$m \times \bar{n}$  matrix such that every column of  $U$  contains one non-zero entry, either a +1 or -1 in row  $i$  if node  $i$  is a source or sink for the commodity. The  $k$ th column of  $A$  corresponds to an arc  $e_k$  of  $G$ ,  $x_k$  can be thought of as the flow in arc  $e_k$ , and  $s_i$  as an exogenous flow either into or out of vertex  $v_i$  depending on whether the coefficient of  $s_i$  in  $U$  is -1 or +1.

Let  $A^0$  denote the matrix  $[A \ U]$ . For a matrix  $B$  consisting of a subset of columns of  $A^0$ , let  $F_B$  be the subgraph of  $G$  consisting of the vertices corresponding to columns of  $U$  and edges corresponding to columns of  $A$  together with vertices incident to such edges.

The main result is the following.

Theorem 3.1 (Johnson [70])

Let  $A^0$  be such that every connected component of  $G$  has at least one vertex corresponding to a column of  $U$ . Then a matrix  $B$  of columns of  $A^0$  is a basis if and only if  $F_B$  is a rooted spanning forest of  $G$ .

This theorem allows one to solve the network linear program graph-theoretically. Such an algorithm is presented in [70]. To illustrate, consider the following constraint matrix  $A^0$ .

$$\begin{array}{cccccccccc}
 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & s_1 & s_3 & s_2 & s_4 \\
 \left[ \begin{array}{cccccccccc}
 1 & 1 & & & & & & 1 & & & \\
 -1 & & -1 & 1 & 1 & & & & & -1 & \\
 & -1 & 1 & -1 & & 1 & & & 1 & & \\
 & & & & -1 & -1 & & & & & -1
 \end{array} \right]
 \end{array}$$

The graph  $G$  is shown in Figure 20. Let the basis  $B$  be

$$B = \begin{array}{c} \begin{array}{cccc} s_1 & s_3 & x_3 & x_6 \end{array} \\ \left[ \begin{array}{cccc} 1 & & & \\ & & -1 & \\ & 1 & 1 & 1 \\ & & & -1 \end{array} \right] \end{array}$$

Then  $F_B$ , a rooted spanning forest is shown in Figure 21.

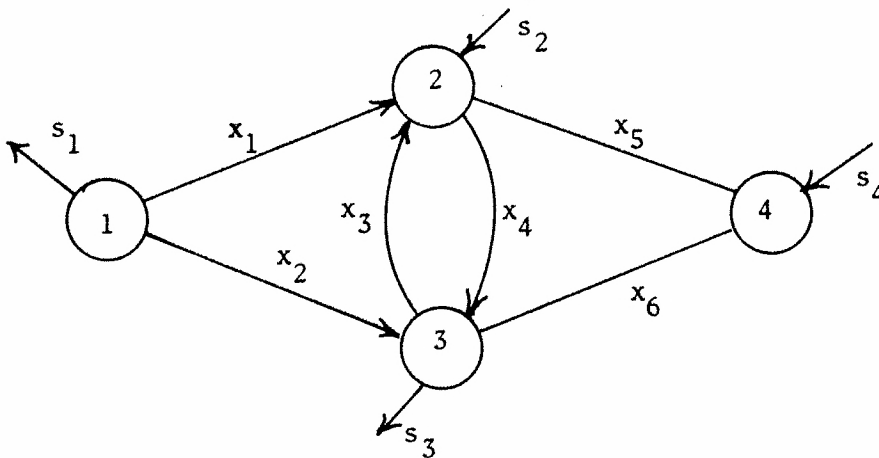


Figure 20. A Linear Programming Network

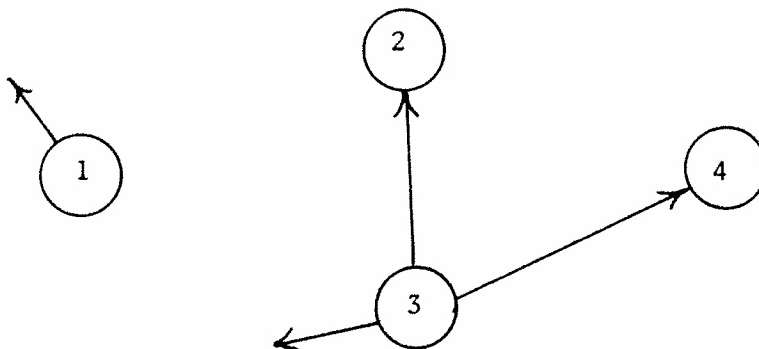


Figure 21. A Rooted Spanning Forest



Solution of Linear Equations by  
Change of Variables

The concepts discussed in this section form the framework for the generalized upper bounding (GUB) procedure of linear programming (Lasdon [81], Chapter 6).

Consider the following system of linear equations

$$\begin{bmatrix} B & C \\ E & F \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{y} \end{bmatrix} = \begin{bmatrix} \underline{d} \\ \underline{a} \end{bmatrix}$$

where  $B$ ,  $C$ ,  $E$ , and  $F$  are matrices, and  $\underline{x}$ ,  $\underline{y}$ ,  $\underline{d}$ , and  $\underline{a}$  are vectors of appropriate consistent dimensions. We shall assume that

$$\begin{bmatrix} B & C \\ E & F \end{bmatrix}$$

is nonsingular. Consider the change of variables

$$\begin{bmatrix} \underline{x} \\ \underline{y} \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 \\ Q_3 & Q_4 \end{bmatrix} \begin{bmatrix} \underline{z} \\ \underline{w} \end{bmatrix}$$

For  $Q_1 = I$ ,  $Q_2 = -B^{-1}C$ ,  $Q_3 = 0$ , and  $Q_4 = I$  we obtain

$$\begin{bmatrix} B & 0 \\ E & F - EB^{-1}C \end{bmatrix} \begin{bmatrix} \underline{z} \\ \underline{w} \end{bmatrix} = \begin{bmatrix} \underline{d} \\ \underline{a} \end{bmatrix}$$

and

$$\begin{bmatrix} \underline{x} \\ \underline{y} \end{bmatrix} = \begin{bmatrix} I & -B^{-1}C \\ 0 & I \end{bmatrix} \begin{bmatrix} \underline{z} \\ \underline{w} \end{bmatrix}$$

Since the matrix  $\begin{bmatrix} I - B^{-1}C \\ 0 & I \end{bmatrix}$  is upper triangular, if  $\underline{z}$  and  $\underline{w}$  are known,

then  $\underline{x}$  and  $\underline{y}$  can easily be found. Solving for  $\underline{z}$  and  $\underline{w}$  we obtain

$$\underline{z} = B^{-1}d$$

$$\underline{w} = (F - EB^{-1}C)^{-1}(\underline{a} - EB^{-1}d)$$

### Application to the Multicommodity Flow Problem

The constraint matrix for the general multicommodity flow problem is

$$A = \begin{bmatrix} A_1 & & & & \\ & A_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & A_r \\ I & I & \dots & I & I \end{bmatrix}$$

Choosing a basis  $\bar{B}$  from  $A$ , we may rearrange the columns and express  $\bar{B}$  as

$$\bar{B} = \left[ \begin{array}{ccc|ccc|c} B_1 & & & R_1 & & & \\ & B_2 & & R_2 & & & \\ & & \ddots & & \ddots & & \\ & & & & & \ddots & \\ & & & & & & R_r \\ \hline & & & & & & \underline{0} \\ \hline & E & & & F & & \end{array} \right]$$

where each  $B_k$  is a basis (rooted spanning forest) for commodity  $k$ ; the columns of  $E$  and  $F$  are unit vectors from the capacity constraints; and  $R_k$  is a subset of columns from  $A_k$  ( $R_k$  may be null). Note that  $\bar{B}$  is of the general form

$$\bar{B} = \begin{bmatrix} B & C \\ E & F \end{bmatrix}$$

Using the technique described in the previous section, we may express  $\bar{B}$ , after a change of variables, as

$$\bar{B}' = \begin{bmatrix} B & 0 \\ E & F - EB^{-1}C \end{bmatrix}$$

Solving the system

$$\bar{B}' \begin{bmatrix} \underline{z} \\ \underline{w} \end{bmatrix} = \begin{bmatrix} \underline{d} \\ \underline{a} \end{bmatrix}$$

we obtain

$$\underline{z} = B^{-1}\underline{d}$$

$$\underline{w} = (F - EB^{-1}C)^{-1}(\underline{a} - EB^{-1}\underline{d})$$

In the multicommodity flow problem,  $B^{-1}$  can be found graph-theoretically and presents no computational problem ( $B^{-1}$  is totally unimodular). The matrix  $(F - EB^{-1}C)$  has a special structure. It can be shown that for the multicommodity flow problem,  $(F - EB^{-1}C)$  can be rearranged to the following form

$$\begin{array}{cc}
 J & U \\
 \left[ \begin{array}{cc} S_1 & 0 \\ S_2 & I \end{array} \right]
 \end{array}$$

Each column in the set  $J$  is the vector expression of a cycle in the network. The rows of  $S_1$  represent those arcs which are saturated, i.e., the capacity constraint is "tight." We see therefore, that a multi-commodity flow basis consists of a set of rooted spanning forests and a set of cycles. The matrix  $(F - EB^{-1}C)$  will be called the cycle matrix of the basis  $\bar{B}$ . To solve the system

$$\bar{B}\underline{x} = \underline{b}$$

we need only explicitly invert  $S_1$  because of the block triangular structure of the cycle matrix and  $\bar{B}'$ . The details are provided in Hartman and Lasdon [45].

#### Non-unimodularity in Multicommodity Flow Bases

The difficulty in solving multicommodity flow problems is that  $S_1$  is generally not totally unimodular. Hence, fractional solutions arise. To illustrate this, consider the example given in Figure 22. This is an MCTP(3,3,2). The arc parameters are (commodity 1 cost, commodity 2 cost, arc capacity). All supplies and demands are two. In the optimal basis are the rooted spanning forests given in Figure 23 and the following cycle matrix

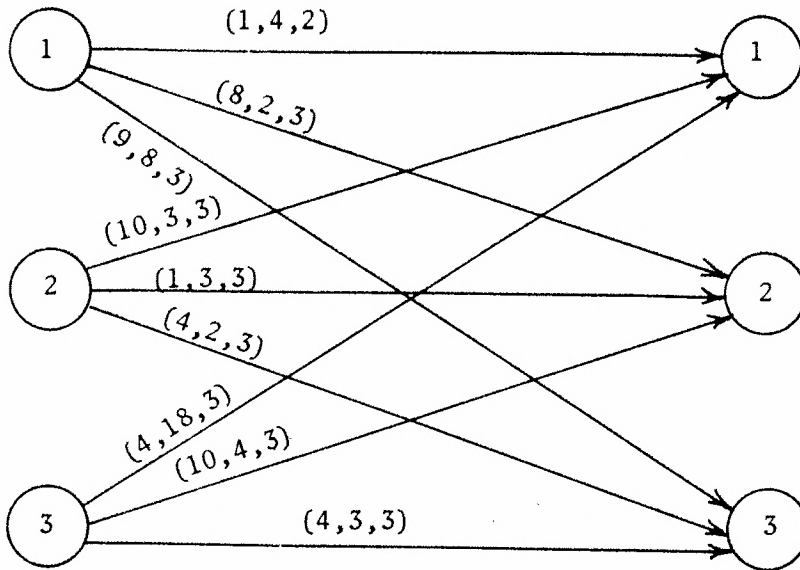


Figure 22. An MCTP(3,3,2) Network

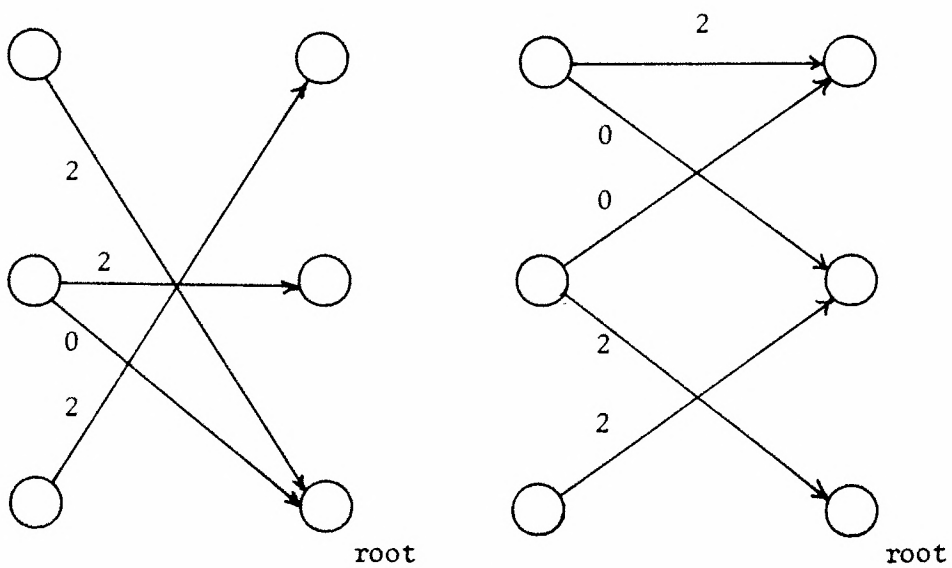


Figure 23. Optimal Rooted Spanning Forests

arc	$x_{11}^1$	$x_{33}^2$	$s_{12}$	$s_{13}$	$s_{21}$	$s_{22}$	$s_{23}$	$s_{31}$	$s_{32}$
1,1	1	-1							
3,3	1	1							
1,2		1	1						
1,3	-1			1					
2,1		1			1				
2,2						1			
2,3		-1					1		
3,1	-1							1	
3,2		-1							1

Note that each column under  $x_{11}^1$  and  $x_{33}^2$  defines a cycle in the network. Arcs (1,1) and (3,3) are saturated. The determinant of  $S_1$  is 2. To find the values of  $x_{11}^1$  and  $x_{33}^2$  we note that arc (1,1) has capacity 2 and arc (3,3) has capacity 3. Since the current flow on arc (1,1) is 2 there is no residual capacity remaining, so we must solve the system

$$\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_{11}^1 \\ x_{33}^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

The solution is  $x_{11}^1 = x_{33}^2 = 3/2$ . If the flows are adjusted around the cycles generated by  $x_{11}^1$  and  $x_{33}^2$  in order to preserve conservation of flow, we obtain the solution given in Figure 24.

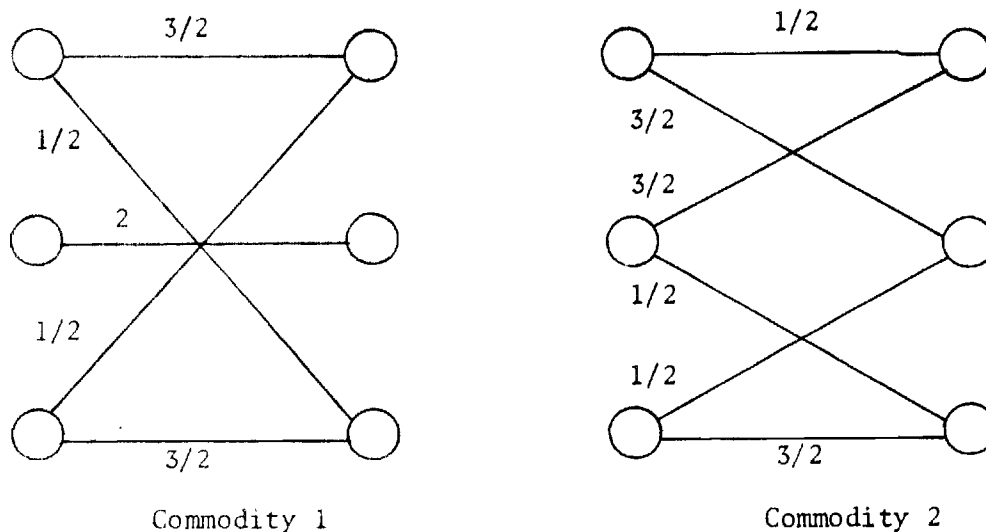


Figure 24. Optimal LP Solution

Theorem 3.2

A necessary and sufficient condition that a multicommodity flow basis yield an integer solution is that the values of the variables  $\underline{y}$  generating cycles are integer, i.e.,

$$\underline{y} = S_1^{-1} \underline{t}$$

be integer, where  $\underline{t}$  is the residual capacity of the arcs corresponding to the rows of  $S_1$ .

Proof. The condition is clearly necessary. To prove sufficiency we note that the solution vector for commodity  $k$  is simply the sum of the solution vector specified by the rooted spanning forest  $B_k$  (which is totally unimodular) plus the values of the arcs  $\underline{y}$  generating basic cycles times their respective columns from the cycle matrix whose elements are all +1, -1, or 0.

We wish to conclude this section with a discussion of the non-unimodular aspect of multicommodity flow problems from a graph-theoretic point of view. Consider the MCTP(3,3,2) in Figure 25 with the flows labelled on the arcs. Assume that arc (1,1) has a capacity of 2 and arc (3,3) has a capacity of 1. We wish to introduce  $x_{11}^1$  into the basis. Let us assume that  $s_{11}$  leaves the basis so that arc (1,1) becomes saturated and generates a second cycle in the basis. For commodity 1, the changes that must occur to the basic variables for introducing  $x_{11}^1$  and conserving flow are shown in Figure 26.

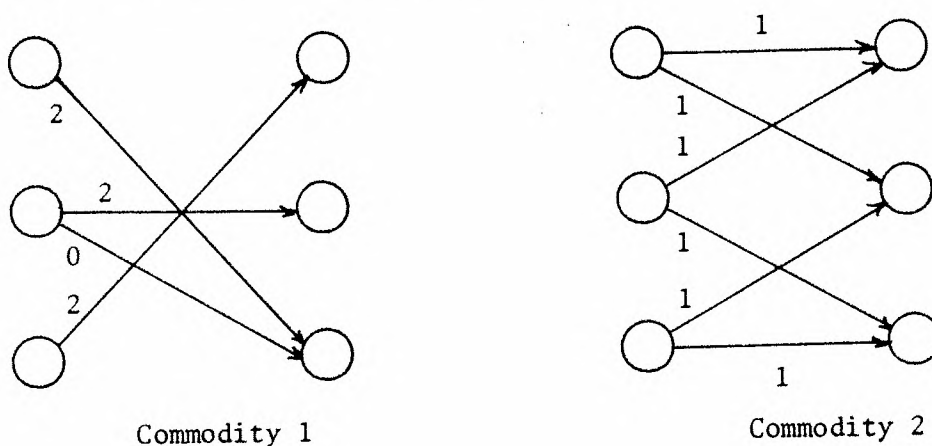


Figure 25. A Two-Commodity Transportation Basis

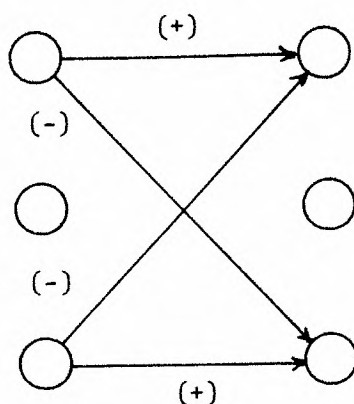


Figure 26. Cycle Flow Exchange for Commodity 1



But now, in order to preserve feasibility on arc (3,3) we must decrease the flow on the cycle formed by  $x_{33}^2$ . This forces a change in the basic variables of commodity 2 as shown in Figure 27. Notice that the flow on  $x_{11}^2$  must also increase proportionally to  $x_{11}^1$ . To satisfy the

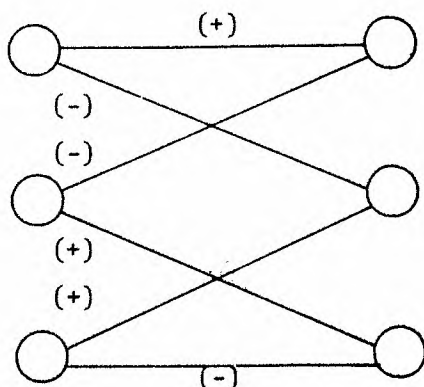


Figure 27. Cycle Flow Exchange for Commodity 2

capacity constraint on arc (1,1), the maximum increase in  $x_{11}^1$  is  $1/2$ , yielding a non-integer basic solution. It is precisely this phenomenon of simultaneous flow change in an arc over several different commodities that causes non-integer solutions and non-unimodularity of  $S_1$ .

### Chain Groups and Cycles in Directed Graphs

We will first review some basic algebraic concepts. A group  $(G, *)$  is an algebraic system defined on a set  $G$  with an operation  $*$  such that

- (i) if  $x, y \in G$ , then  $x*y \in G$
- (ii)  $*$  is associative
- (iii) there exists  $e \in G$  such that if  $g \in G$ ,  $e*g = g$
- (iv) for every  $g \in G$ , there exists  $h \in G$  such that  $h*g = e$

A group is Abelian if for all  $g, h \in G$ ,  $g*h = h*g$ . A ring is a two-

operation system  $(R, +, \cdot)$  with operations  $+$  and  $\cdot$  such that

- (i)  $(R, +)$  is an abelian group
- (ii)  $\cdot$  is associative in  $R$
- (iii)  $\cdot$  is distributive over  $+$

The integers are a ring with  $+$  and  $\cdot$  being the usual addition and multiplication. A ring  $(A, +, \cdot)$  is commutative if and only if  $\cdot$  is commutative in  $A$ . It is a ring with identity if and only if  $A$  contains an identity for the operation  $\cdot$ . An element  $a \neq 0$  in a ring is called a divisor of zero if and only if there exists a  $b \neq 0$  in  $A$  such that  $a \cdot b = 0$  or  $b \cdot a = 0$ . A field is a commutative ring with identity in which every nonzero element has a multiplicative inverse.

Much of the following discussion is taken from Tutte [130].

Let  $R$  be a commutative ring with identity and no divisors of zero. In our applications,  $R$  will be the integers or the field of residues mod 2,  $GF(2)$ . A chain on a finite set  $M$  is a mapping  $f$  of  $M$  into  $R$ . If  $a \in M$ , then  $f(a)$  is the coefficient of  $a$  in the chain  $f$ . The domain  $|f|$  of  $f$  is the set of all  $a \in M$  such that  $f(a) \neq 0$ . If  $|f| = \emptyset$ , then  $f$  is the zero chain on  $M$ . The sum  $f+g$  of two chains on  $M$  is a chain on  $M$  defined by

$$(f+g)(a) = f(a) + g(a)$$

Thus, the chains on  $M$  are the elements of an additive abelian group  $A(M)$ . A chain group on  $M$  is any subgroup of  $A(M)$ .

Let  $N$  be a chain group on  $M$ . A chain  $f \in N$  is an elementary chain of  $N$  if it is non-zero and there is no non-zero chain  $g \in N$  such that  $|g|$  is a proper subset of  $|f|$ . A primitive chain is an

elementary chain whose coefficients are restricted to the values 0, 1, and -1. A chain group is regular if for each elementary chain  $f$  of  $N$  there exists a primitive  $g \in N$  such that  $|g| = |f|$ . This concept is closely related to regular matroids discussed in Chapter II; in fact, Tutte's development of matroid theory stems from consideration of chain groups [126].

A chain group that is important in our applications derives from linear graphs. If  $G$  is a directed graph, we call an oriented cycle of  $G$  a cycle in which all arcs are pointed in the same direction as one traverses the cycle. It can be shown that the oriented cycles of  $G$  over  $R$  constitute a chain group  $\Gamma_R(G)$ . If  $R$  is the ring of integers,  $I$ , then the oriented cycles of  $G$  constitute a chain group  $\Gamma_I(G)$ .

Theorem 3.3 (Tutte [130])

$\Gamma_I(G)$  is a regular chain group.

The properties of regular matroids are derived from regular chain groups. For any chain  $f$  on a set  $E$  over  $R$ , define  $[f(e_1), f(e_2), \dots, f(e_n)]$  to be the representative vector of  $f$ , i.e., the vector of coefficients of the chain  $f$ . If  $f \in \Gamma_I(G)$ , and is primitive, then  $f(e_i)$  is 0, +1, or -1, and the representative vector is simply the vector expression of the cycle in  $G$ . Let  $K$  be a matrix whose rows are the representative vectors of all oriented cycles in  $G$ , called the representative matrix. The important result concerning representative matrices of regular chain groups is that they are totally unimodular. The detailed theoretical development can be found in Tutte [130]. We shall use the

concepts of chains and chain groups in the context of the multicommodity flow problem.

### Chains and Cycles in Multicommodity Networks

As we have seen, non-unimodularity of the cycle matrix leads to non-integer solutions in multicommodity network flow problems. The non-unimodularity arises from the fact that cycles in the basis do not form a fundamental set of cycles relative to the same spanning forest. A fundamental cycle (relative to a forest  $F$ ) is a cycle formed when an out-of-forest arc is added to  $F$ . The out-of-forest arcs are commonly called chords. The set of cycles formed by adding chords to  $F$  one at a time is called a fundamental set of cycles and can be shown to form a basis for the cycle subspace, i.e., any other cycle is a linear combination of the fundamental set. A fundamental cycle matrix is the matrix of the vector representations (chains) of a fundamental set of cycles. It is well known (Ponstein [97]) that a fundamental cycle matrix is totally unimodular. However, a set of independent cycles that are fundamental relative to different forests may not have a totally unimodular matrix representation. These are precisely the types of cycles that appear in multicommodity network flow bases.

We have previously noted that the set of oriented cycles of a graph  $G$  forms a regular chain group whose representative matrix is totally unimodular. Let  $G$  be any directed graph. Define  $H$  as a matrix whose rows are the representative vectors of primitive chains of all elementary cycles and edge-disjoint unions of cycles of  $G$ . The coefficients of each chain are determined by giving each cycle

an arbitrary orientation; that is, if we begin at some vertex and transverse a path through the cycle, the coefficient of edge  $e_j$  is  $f(e_j) = +1$  if  $e_j$  is a forward arc, and  $f(e_j) = -1$  if  $e_j$  is a reverse arc. Let  $H^*$  be the matrix  $H$  with all  $-1$ 's replaced by  $+1$ . Each row of  $H^*$  is a vector representation of the domain of the cycle. Let  $N(H)$  be the set of chains corresponding to the rows of  $H$ . Note that each member of  $N(H)$  is a primitive chain.

Consider the domains of the chains of  $N(H)$  represented by  $H^*$ . To avoid ambiguity, we call the domain of a cycle a circuit. The following are well known in graph theory.

Theorem 3.4 (Liu [85])

The mod 2 sum of any two rows of  $H^*$  is a row of  $H^*$ .

Theorem 3.5 (Liu [85])

The set of rows of  $H^*$  is an abelian group under modulo 2 summation.

An equivalent statement of Theorem 3.5 is that the set of domains of all cycles and edge-disjoint unions of cycles in a graph form an abelian group under the set operation  $\oplus$  where  $\oplus$  is defined by

$$X \oplus Y = (X \cup Y) - (X \cap Y)$$

A binary operation on a set  $A$  is any function from  $A \times A$  to  $A$ . Let us define a binary operation  $*$  as follows. Let  $f \in N(H)$ ,  $g \in N(H)$  and  $e$  be any member of  $|f| \cap |g|$ . If  $|f| \cap |g| = \emptyset$ , then define  $f * g = f + g$  where  $+$  is the usual addition operator. If  $e \in |f| \cap |g|$  and  $f(e) = +1$  and  $g(e) = -1$  or vice-versa, then  $f * g = f + g$ . If  $f(e) = g(e) = +1$  or  $f(e) = g(e) = -1$ , then  $f * g = f + (-g)$ .

The operation  $*$  may not be well-defined on the set  $N(H)$  of a graph  $G$ . Consider the graph in Figure 28. Let  $f = e_1 - e_2 + e_4$  and  $g = e_3 + e_5 + e_2$ .  $|f| \cap |g| = \{e_2\}$ , then  $f * g = e_1 + e_3 + e_4 + e_5$  and  $f * g \in N(H)$ . However, let  $f = -e_1 - e_6 + e_5 + e_2$  and  $g = e_1 + e_3 + e_4 + e_5$ . Then  $|f| \cap |g| = \{e_1, e_5\}$ . Defining  $f * g$  with respect to  $e_1$  we obtain  $f * g = e_3 - e_6 + 2e_5 + e_2 + e_4 \notin N(H)$ . With respect to  $e_5$  we obtain  $f * g = -2e_1 - e_6 - e_3 + e_2 - e_4 \notin N(H)$ . Neither of these chains are primitive nor multiples of primitive chains. Therefore  $*$  is not well-defined on the set  $N(H)$ .

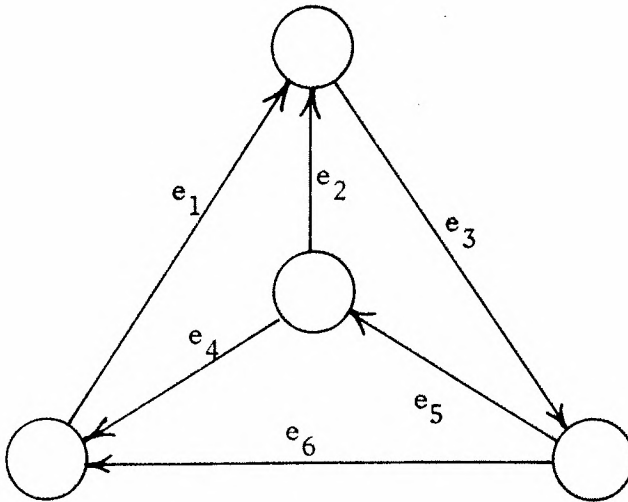


Figure 28. A Directed Graph

### Theorem 3.6

Let  $P \subseteq N(H)$ . If  $*$  is well-defined on the set  $P$ , then  $(P, *)$  is a group.

Proof. Let  $f, g, k \in P$ . Since  $f * g = h \in P$  and  $g * k = \ell \in P$ , it follows that

$$(f * g) * k = \underline{+} f \underline{+} g \underline{+} k = f * (g * k)$$



since the operation  $*$  is simply the addition or subtraction of real vectors which clearly associative. The zero chain is the identity element, and  $f^{-1} = (-f)$  since  $f*(-f) = 0$ . Therefore, inverses exist and the theorem is complete.

It immediately follows that if  $(P,*)$  is a group, then it is a regular chain group since all chains are primitive by nature of  $*$ .

We may determine whether or not  $*$  is well-defined graph-theoretically. Let  $f$  and  $g$  be two chains corresponding to cycles in a graph,  $G$ . Give each cycle an arbitrary orientation on  $G$  and assign to each arc either a  $(+)$  or  $(-)$  depending on whether the arc is a forward or reverse arc with respect to the given orientation. Define  $E = \{e_j | e_j \in |f| \cap |g|\}$ . Then if every member of  $E$  has a  $(+)$  sign in both cycles, a  $(-)$  sign in both cycles, or opposite signs in both cycles, then  $f*g$  is defined and a primitive cycle. For example, in the graph in Figure 30, let  $f$  and  $g$  be represented by the following two cycles.

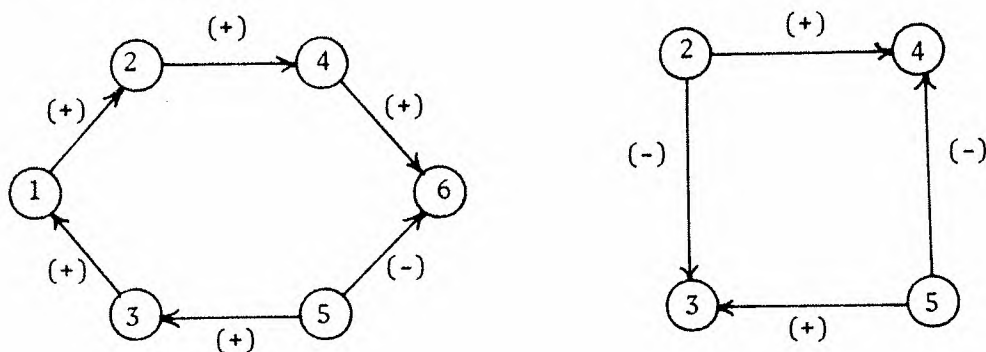


Figure 29. Two Directed Cycles

Then  $E = \{e_4, e_7\}$  and in this case  $f*g$  is defined. This graph-theoretic condition is simply an interpretation of the definition of  $*$ .

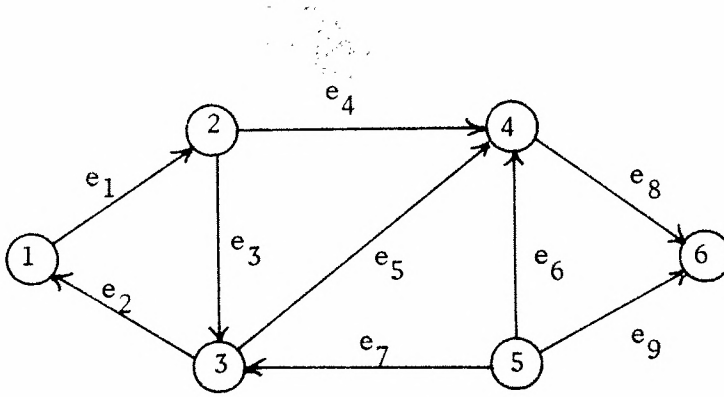


Figure 30. A Directed Graph

A set of cycles  $P$  generates a group of cycles  $\bar{P}$  under  $\oplus$  if orientations on the arcs are ignored. We first consider the domains of cycles in  $P$ , generate the domains of cycles in  $\bar{P}$ , and finally let  $\bar{P}$  be the directed cycles obtained. For example, let  $G$  be as in Figure 30, and let

$$f_1 = e_6 + e_8 - e_9$$

$$f_2 = e_5 - e_6 + e_7$$

$$f_3 = e_4 - e_5 + e_3$$

$P = \{f_1, f_2, f_3\}$ . We obtain

$$|f_1| \oplus |f_2| \Rightarrow f_4 = e_5 + e_8 - e_9 + e_7$$

$$|f_1| \oplus |f_3| \Rightarrow f_5 = e_6 + e_8 - e_9 + e_4 - e_5 - e_3$$

$$|f_2| \oplus |f_3| \Rightarrow f_6 = e_4 - e_6 + e_7 - e_3$$

$$|f_1| \oplus |f_2| \oplus |f_3| \Rightarrow f_7 = e_4 + e_8 - e_9 + e_7 - e_3$$



Then  $\bar{P} = \{f_1, f_2, \dots, f_7\}$ .  $\bar{P}$  is clearly a subset of  $N(H)$ . We need only check that  $f_i * f_j$  is defined for  $f_i, f_j \in \bar{P}$ . The reason we must find  $\bar{P}$  is encompassed in the following.

### Theorem 3.7

Let  $P$  be a set of cycles in a basis  $B$  to a multicommodity flow problem. If  $(\bar{P}, *)$  is a group, then the solution  $\underline{x}_B = B^{-1} \underline{b}$  is integer.

This follows from the remarks concerning regular chain groups. We cannot simply check every pair of cycles in  $P$ . For example, consider the following submatrix for some set of cycles  $P$ .

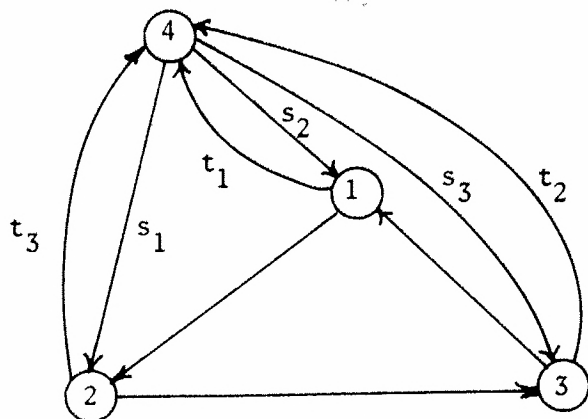
$$\begin{array}{ccc} f_1 & f_2 & f_3 \\ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \end{array}$$

Every pair of columns has the property that  $f_i * f_j$  is defined; however,  $(f_1 * f_2) * f_3$  is not, and the determinant of the above matrix is 2.

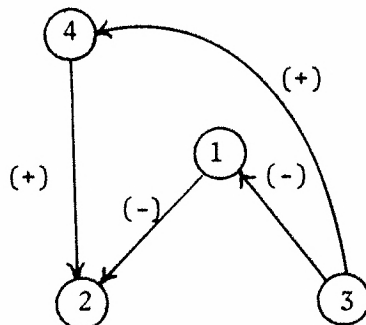
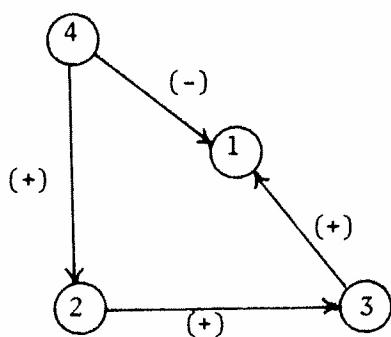
### Corollary 3.7.1

If  $G$  is the graph of a multicommodity flow problem such that every cycle in  $G$  has the property that  $f_i * f_j$  is defined, then all basic solutions are integer.

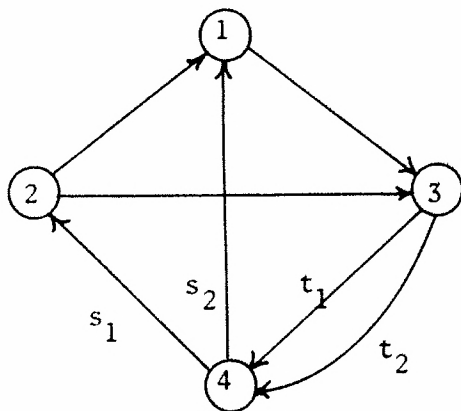
This corollary can provide a relatively simple method of determining whether or not a multicommodity flow problem will be integer. For example, in Ford and Fulkerson's three-commodity example, we have the following network.



Two cycles which do not meet this condition are



An example where the condition is satisfied is the following



We have shown that a basic solution to a multicommodity flow problem will be integer for any right hand side if the cycles in the basis generate a group under  $*$ . This is not a necessary condition; for example,  $S_1$  may be unimodular if the basic cycles do not form a group under  $*$  but the operation fails for some non-saturated arc.

#### An Alternate Proof of the MCTP Unimodularity Theorem

The results of the previous section can be used to provide an exceedingly simple proof of sufficiency of Theorem 2.13. All that must be shown is that  $f*g$  is defined for all cycles in the  $MCTP(2,n,r)$ . The proof using matroid theory is more appealing to the author because of the matroid graphs that result.

Consider a two-source,  $n$ -sink transportation network in Figure 31. We will denote such a graph by  $K(2,n)$ .

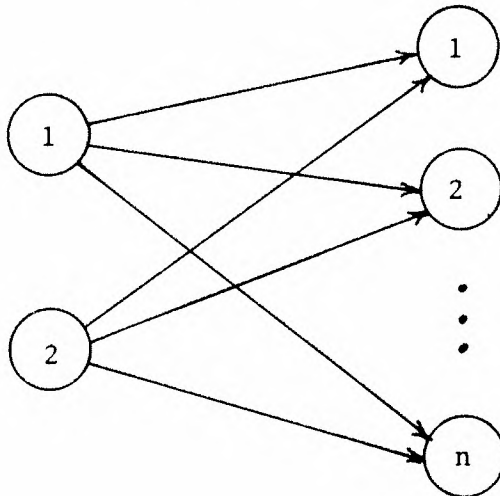


Figure 31. A Two-Source,  $n$ -Sink Transportation Network

#### Lemma 3.10

Any cycle of  $K(2,n)$  has exactly four arcs.

Proof. Obvious.

Lemma 3.11

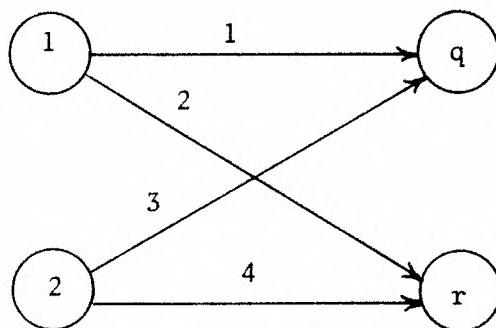
If  $f$  and  $g$  are any two independent cycles (chains) of  $K(2,n)$ , then

$p = |f| \cap |g|$  is 0, 1, or 2.

Proof. By Lemma 3.10, if  $p = 4$  then  $f = g$  and hence not independent.

Since any set of three arcs that is a subset of a cycle of  $K(2,n)$  forms a tree on the subset of incident nodes of  $K(2,n)$ , there is a unique arc that completes the cycle. Hence either  $f = g$  or,  $f$  or  $g$  is not a cycle. Q.E.D.

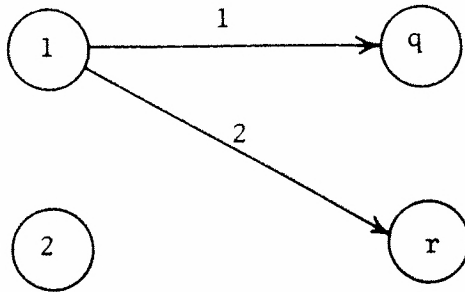
To complete the proof of sufficiency of the unimodularity theorem, note that if  $p = 0$  or 1, then  $f * g$  is clearly defined. We need only consider the case when  $p = 2$ . Let  $f$  be any cycle of  $K(2,n)$ , as shown below.



If  $p = 2$ , then any cycle  $g$  contains one of the following pairs of arcs:

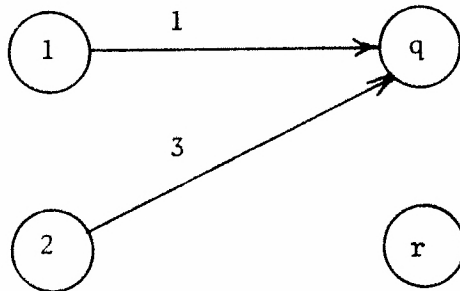
$(1,2)$ ,  $(1,3)$ ,  $(1,4)$ ,  $(2,3)$ ,  $(2,4)$ ,  $(3,4)$ . Elements within the sets  $\{(1,2), (3,4)\}$ ,  $\{(1,3), (2,4)\}$ , and  $\{(1,4), (2,3)\}$  are topologically indistinguishable. We consider each set in turn.

Case 1. Common arcs are (1,2)

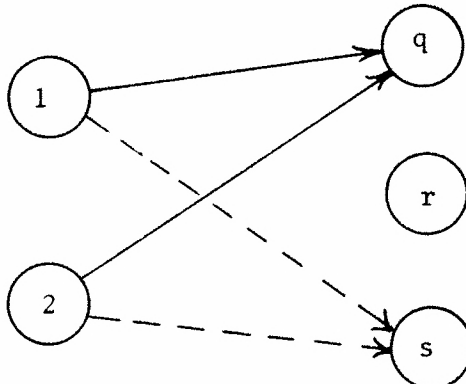


In this case there is only one set of arcs, namely (3,4), that form a cycle and then  $f = g$ .

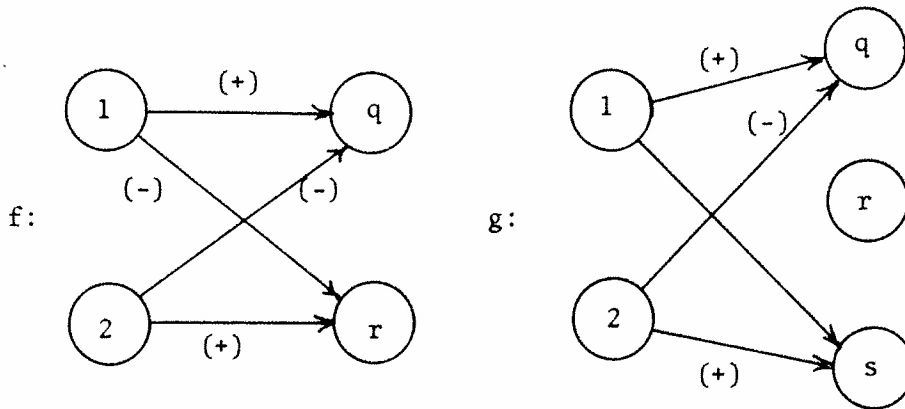
Case 2. Common arcs are (1,3)



If a cycle is formed by arcs 3 and 4 then  $f = g$ . If not, there is some node  $s$  that forms the cycle  $g$ .

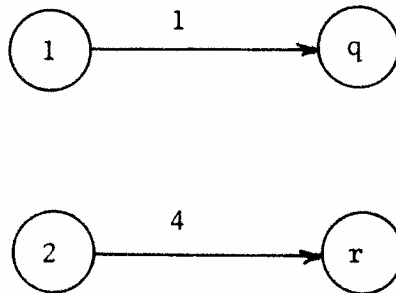


Giving  $f$  and  $g$  an arbitrary orientation yields



Clearly  $f \star g$  is defined.

Case 3. Common arcs are  $(1,4)$



As in case 1 there is only one unique cycle that can be completed and then  $f = g$ . This completes the proof.

## CHAPTER IV

### COMBINATORIAL COMPLEXITY OF INTEGER MULTICOMMODITY TRANSPORTATION PROBLEMS

#### Introduction

Edmonds [21, 23] has successfully solved certain classes of combinatorial optimization problems on graphs. By "successfully solved" we mean there exists an algorithm whose number of computations is bounded by a polynomial function of the input parameters, such as number of nodes, number of arcs, etc. Such an algorithm is called a "good" or polynomially bounded algorithm. In the non-unimodular matching problem, Edmonds [21] derived a class of cutting planes which yielded a polyhedron all of whose vertices have integer coordinates. Because of the graph-theoretic structure of the MCTP, one wonders whether or not such a class of cuts exists for the MCTP that can be a priori specified as in the general matching problem. In the current state-of-the-art, there do not exist polynomially bounded algorithms for such problems as the travelling salesman problem. Karp [72] has defined an entire class of such problems, called "polynomial complete" problems, where if a polynomially bounded algorithm exists for one of them, it exists for all of them. In this chapter we will investigate this question for the MCTP.

#### The Multicommodity Assignment Problem

To simplify our discussion, we will consider a special case of

the MCTP when all supplies and demands are equal to one. Without loss of generality, we will assume that the number of sources equals the number of sinks for the multicommodity assignment problem (MCAP). The MCAP can be formulated as follows.

$$\text{MCAP:} \quad \text{Minimize} \quad z = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^r c_{ij}^k x_{ij}^k$$

$$\text{subject to} \quad \sum_{i=1}^n x_{ij}^k = 1 \quad \text{for all } k, j \quad (4.1)$$

$$\sum_{j=1}^n x_{ij}^k = 1 \quad \text{for all } k, i \quad (4.2)$$

$$\sum_{k=1}^r x_{ij}^k \leq u_{ij} \quad \text{for all } i, j \quad (4.3)$$

$$x_{ij}^k = 0, 1 \quad \text{for all } i, j, k \quad (4.4)$$

We wish to note that the MCAP is distinct from the "multidimensional assignment problem" (MDAP) that has appeared in the literature (Pierskalla [95]). The MDAP for  $p \leq q \leq r$  is stated as

$$\text{MDAP:} \quad \text{Minimize} \quad \sum_{i=1}^p \sum_{j=1}^q \sum_{k=1}^r c_{ijk} x_{ijk}$$

$$\text{subject to} \quad \sum_i \sum_j x_{ijk} \leq 1 \quad \text{for all } k$$

$$\sum_i \sum_k x_{ijk} \leq 1 \quad \text{for all } j$$

$$\sum_j \sum_k x_{ijk} = 1 \quad \text{for all } i$$

$$x_{ijk} = 0, 1 \quad \text{for all } i, j, k$$



Intuitively, the MDAP can be thought of as finding a matching on a three-dimensional lattice of points.

By ignoring the constraint (4.4) in the MCAP, we obtain the relaxed linear program  $\overline{\text{MCAP}}$ . Let  $V(\overline{\text{MCAP}})$  be the set of vertices of the constraint set of  $\overline{\text{MCAP}}$ . Let  $V(\overline{\text{MCAP}}) = I(\overline{\text{MCAP}}) \cup N(\overline{\text{MCAP}})$  where  $I(\overline{\text{MCAP}})$  is the set of vertices having all integer coordinates, and  $N(\overline{\text{MCAP}})$  the set of vertices some of whose coordinates are not all integer. Let  $S(\overline{\text{MCAP}})$  be the set of all feasible solutions to the MCAP. Such a solution will be called a multicommodity matching,  $M$ . Let  $M_1, M_2, \dots, M_r$  be matchings (or equivalently basic feasible solutions) to the associated single commodity assignment problems.  $M$  is clearly a union of single commodity matchings; that is, every multicommodity matching can be expressed as

$$M = M_1 \cup M_2 \cup \dots \cup M_r$$

Let  $T = \{M \mid M = M_1 \cup M_2 \cup \dots \cup M_r\}$ . Then clearly  $S(\text{MCAP}) \subseteq T$  and

$$S(\text{MCAP}) = \{M \in T \mid (4.3) \text{ is satisfied}\}$$

#### Theorem 4.1

$$M \in S(\text{MCAP}) \iff M \in I(\overline{\text{MCAP}})$$

Proof. Let  $M \in S(\text{MCAP})$ . Then  $M = M_1 \cup M_2 \cup \dots \cup M_r$ .

The constraint matrix for  $\overline{\text{MCAP}}$  has the form

$$A = \begin{bmatrix} A_1 & & & & \\ & A_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & A_r \\ I & I & \dots & I & I \end{bmatrix}$$

Each  $M_k$  represents a basic feasible solution to  $A_k \underline{x}^k = \underline{1}$  with basis  $B_k$ . But also, (4.3) is satisfied, so the slack variables are non-negative, i.e.,  $\underline{s} = \underline{u} - \sum_k \underline{x}^k \geq 0$ . Let  $R_k$  be the matrix of the last  $n^2$  rows of  $A$  corresponding to the columns specified by  $B_k$ . Then  $M$  is represented by the following basis

$$B = \begin{bmatrix} B_1 & & & & \\ & B_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & B_r \\ R_1 & R_2 & \dots & R_r & I \end{bmatrix}$$

Since all slack variables are basic and  $\{B_k\}$  is totally unimodular, we are done. To prove the converse, we note that from constraints (4.1) and (4.2) and  $x_{ij}^k \geq 0$ , no  $x_{ij}^k$  can be greater than one. Hence if  $M \in I(\overline{\text{MCAP}})$  since (4.1) and (4.2) are satisfied then  $M$  is clearly a member of  $S(\text{MCAP})$ .

#### Corollary 4.1.1

$\overline{\text{MCAP}}$  has no interior lattice points.

We note that the multicommodity assignment problem is highly degenerate; thus there may be several different basis representations corresponding to the same vertex. This would make extreme point solution procedures more difficult to develop.

As we have seen in Chapter III, a multicommodity flow basis consists of a set of spanning trees and a set of cycles. In a feasible solution to the MCAP, every cycle must be 0 or 1. In view of Theorem 4.1, for any basis containing integer basic cycles, there exists another basis containing no cycles and corresponding to the same vertex.

At this point we wish to show by counterexample that the optimal integer solution to the MCTP does not necessarily occur at an extreme point. (The example given in Chapter III actually has its optimal integer solution at an extreme point.) Consider the four-commodity problem with costs given in Figure 32. All supplies and demands are 2 and the vector of arc capacities is  $[2, 5, 5, 4, 3, 4, 4, 4, 3]$ . The optimal integer solution is

$$\underline{x}^1 = [1, 0, 1, 0, 2, 0, 1, 0, 1]$$

$$\underline{x}^2 = [1, 1, 0, 1, 0, 1, 0, 1, 1]$$

$$\underline{x}^3 = [0, 1, 1, 2, 0, 0, 0, 1, 1]$$

$$\underline{x}^4 = [0, 1, 1, 1, 0, 1, 1, 1, 0]$$

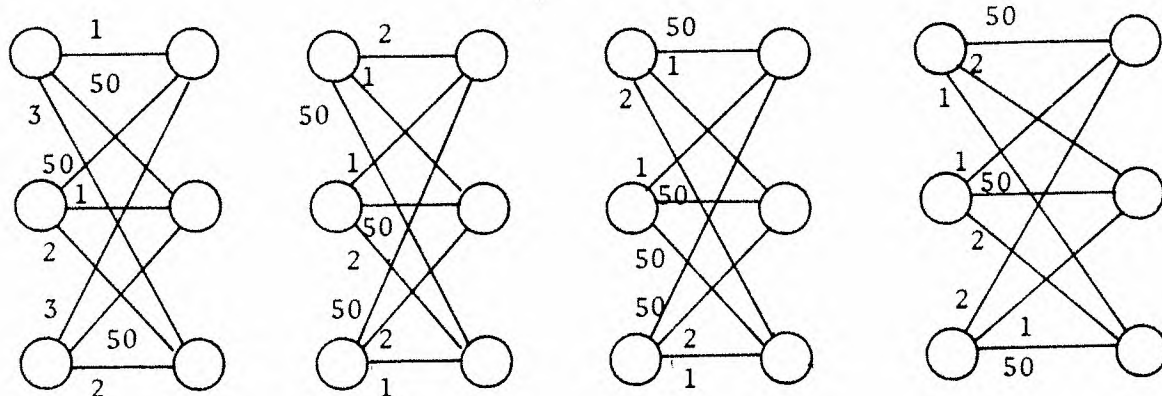
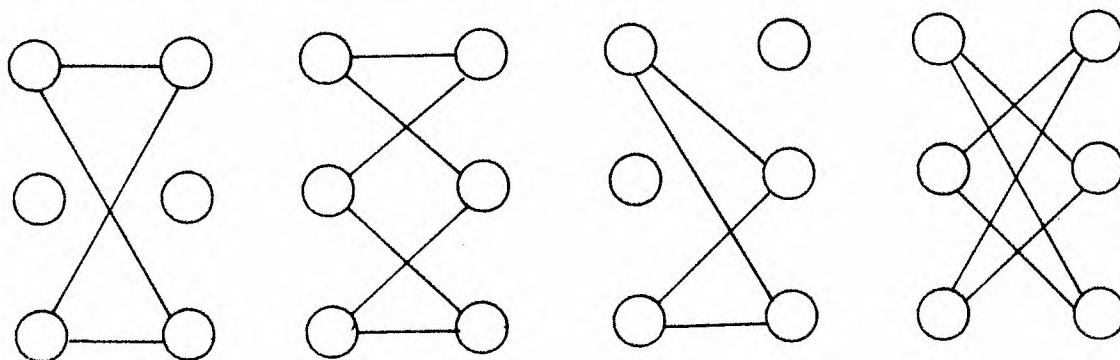


Figure 32. An MCTP(3,3,4)

This is not an extreme point solution since the following cycles have positive flow but are linearly dependent.



### Matroid Theory and Combinatorial Optimization

Matroid theory was introduced in Chapter II as a tool in proving the unimodularity theorem for the multicommodity transportation problem. There are basically two schools of thought with regard to matroid theory: Tutte's development in terms of regular abelian chain groups; and the application of matroids in combinatorial optimization problems as explored by Edmonds [20, 21, 23, 24], Karp [72], and Lawler [82, 83, 84] to name a few. In this section we shall discuss the latter and develop an interesting result concerning the multicommodity assignment

and transportation problems. (Much of the material in this section is taken from Lawler [84]).

There are several equivalent axiom systems which characterize a matroid (cf. Harary and Welsh [44]). The axioms C1 and C2 presented in Chapter II are commonly referred to as circuit axioms. We shall now present a second definition of a matroid. A matroid  $M = (E, J)$  is a structure defined on a finite set  $E$  of elements where  $J$  is a non-empty family of subsets of  $E$  (called independent sets) satisfying

Axiom I1 If  $I \in J$  and  $I' \subseteq I$  then  $I' \in J$ .

Axiom I2 If  $I_p$  and  $I_{p+1}$  are sets in  $J$  containing  $p$  and  $p+1$  elements respectively, then there exists an element  $e \in I_{p+1} - I_p$  such that  $I_p + e \in J$ .

These axioms are called the independence axioms.

A set which is not independent is said to be dependent. A minimal dependent set is called a circuit. A fundamental theorem of matroid theory is that if  $I$  is independent and  $I + e$  (i.e.,  $I \cup \{e\}$ ) is dependent, then  $I + e$  contains precisely one circuit.

We shall now present some examples of matroids (Lawler [82,84]).

- (i) Let  $E$  be the columns of an  $m \times n$  matrix  $C$  and let  $J$  be the family of linearly independent subsets of columns. The matroid  $M = (E, J)$  is said to be the matroid of the matrix  $C$ ; such a matroid is said to be matric.
- (ii) Let  $E$  be the set of arcs of a linear graph  $G$ , and let  $J$  be the family of subsets of arcs which contain no cycles. The matroid  $M = (E, J)$  is called a graph-matroid. Every

graph matroid is matric, as seen by considering the node-arc incidence matrix of  $G$  over the field  $GF(2)$ .

- (iii) Let  $P = \{p_i; i = 1, 2, \dots, m\}$  be a partition of the set  $E$  into  $m$  blocks or equivalence classes. Let  $d_1, d_2, \dots, d_m$  be non-negative integers. Let  $J$  be the family of all subsets  $I$  of  $E$  such that

$$|I \cap p_i| \leq d_i \quad i = 1, 2, \dots, m$$

Then  $M = (E, J)$  is called a partition matroid.

Example (ii) is particularly interesting. If we assign a numerical weight to each member of  $E$  and consider the problem of finding a maximally weighted (or minimally weighted) maximal independent subset (basis) of  $E$ , a very simple algorithm, namely, the greedy algorithm, solves the problem. A well-known example of this is the minimal spanning tree problem. A maximal independent subset of  $E$  is a spanning tree. Furthermore, the number of computations necessary to solve the problem is polynomially bounded, in this case by the number of arcs in the graph. In fact, any optimization problem in which the feasible solutions are the bases of a matroid can be solved by the greedy algorithm. Jack Edmonds [20] provides an elegant discussion of this fact.

Let  $M_1 = (E, J_1)$  and  $M_2 = (E, J_2)$  be two given matroids. A subset  $I \in J_1 \cap J_2$  is said to be a matroid intersection of  $M_1$  and  $M_2$ . We give some examples of matroid intersections.

- (iv) Let  $C$  be an  $m \times n$  matrix. Suppose we draw a horizontal line through  $C$  so that there are  $M_1$  rows above the line

and  $M_2$  below. We can speak of a subset of the columns as being linearly independent both "above the line" and "below the line." Any such subset of columns is a matroid intersection.

- (v) Suppose two graphs  $G_1$  and  $G_2$  are assembled from the same set of arcs  $E$ . A subset  $I \subseteq E$  is a matroid intersection if it is cycle-free in both  $G_1$  and  $G_2$ .
- (vi) Let  $G$  be a bipartite graph in which each arc extends between a node in a set  $S$  and a node in a set  $T$ . A matching in  $G$  is a subset of edges, no two of which meet at the same vertex. Let  $M_1$  be a partition matroid which has as its independent sets all subsets of arcs, no two of which meet at the same node of  $S$ . Let  $M_2$  be a partition matroid which has as its independent sets all subsets of arcs, no two of which meet at the same node of  $T$ . Every matching is an intersection of matroids  $M_1$  and  $M_2$ , and vice-versa.

Example (vi) characterizes the optimal assignment problem. We may also characterize example (vi) in terms of example (iv) by considering the node-arc incidence matrix of  $G$  where rows "above the line" represent source nodes, and rows "below the line" represent sink nodes.

Two matroids of an intersection problem do not have to be of the same type.

- (vii) Let  $G$  be a directed graph. Let  $M_1$  be the graph-matroid of  $G$  (for which arc orientations are irrelevant). Let  $M_2$  be a partition matroid which has as its independent



sets all subsets of arcs, no two of which are directed into the same node. An intersection of these two matroids is a union of directed trees rooted from a point, commonly called a branching.

In all these examples of two-matroid intersection problems, if we assign weights to the members of  $E$  then efficient, in the sense of polynomially bounded, algorithms exist. Edmonds has developed algorithms for problems (vi) and (vii), [21, 23]. Lawler [84] provides a general format. Edmonds [20] shows that polynomially bounded algorithms exist for all two-matroid intersection problems.

For example (vii) we may also define a matroid  $M_3$  which has as its independent sets all subsets of arcs, no two of which are directed out of the same node. One can then consider the problem of finding a maximum weight set of arcs that are independent in all three matroids. This is precisely the travelling salesman problem. There are no known polynomially bounded algorithms for three-matroid intersection problems; if there were then one could well solve the travelling salesman problem, the multidimensional assignment problem, and many other combinatorial optimization problems (Karp [72]). Lawler [84] has shown the following.

Theorem 4.2 (Lawler [84])

There exists a polynomially bounded algorithm for the intersection of three matroids if and only if there exists a polynomially bounded algorithm for the intersection of an arbitrary number ( $\geq 4$ ) of matroids.



Application to the Multicommodity Assignment  
and Transportation Problems

In this section we wish to show that the integer multicommodity assignment and transportation problems belong to the class of three-matroid intersection problems, thus apparently precluding the search for polynomially bounded algorithms, at least with the current state-of-the-art.

Theorem 4.3

The integer multicommodity assignment problem is a three-matroid intersection problem.

Proof. Structure the constraint matrix for the MCAP (assuming without loss of generality that  $m = n$ ) so that the first  $n$  rows are the source node constraints for all commodities, the next  $n$  rows are the sink node constraints, and the last  $n^2$  rows are the capacity constraints. Denote these sets of rows as 1, 2, and 3. In the spirit of examples (iv) and (vi), let  $M_1$  and  $M_2$  be partition matroids whose independent sets are linearly independent columns in row sets 1 and 2 respectively. These have exactly the same graph-theoretic interpretation as in example (vi) except that we are dealing with multiple commodities. Now define  $M_3$  as follows using the notation in example (iii). Let  $P_i$  be the set of columns that correspond to arc  $i$  in the MCAP. (For an  $r$ -commodity problem,  $|P_i| = r$  for all  $i$ ). Let  $d_i$  be the capacity of arc  $i$ . Let  $J_3$  be the family of subsets  $I$  of  $E$  such that

$$|I \cap P_i| \leq d_i \quad \text{for all } i$$

Then  $M_3 = (E, J_3)$  is a partition matroid. A set of arcs that is a feasible solution to the MCAP must be independent in  $M_1$ ,  $M_2$ , and  $M_3$ .

We now show that the MCAP cannot be formulated as a two-matroid problem. Since Edmonds has shown that the assignment problem  $(M_1, M_2)$  is a two-matroid problem, we will show that  $(M_1, M_3)$  and  $(M_2, M_3)$  cannot be one-matroid problems thus completing the proof. This follows from the fact that the rows of  $(M_1, M_3)$  or  $(M_2, M_3)$  can be partitioned into disjoint sets corresponding to  $M_1$  and  $M_3$ , or to  $M_2$  and  $M_3$  such that each column has exactly one 1 in the first row set and one 1 in the second row set. But this is simply another assignment problem with degree constraints that are not necessarily one on the nodes corresponding to row set 3 since we have the incidence matrix of a bipartite graph. Hence  $(M_1, M_3)$  and  $(M_2, M_3)$  are two-matroid problems and therefore  $(M_1, M_2, M_3)$  is a three-matroid intersection problem.

For illustrative purposes, consider  $\text{MCAP}(3,3,2)$ . The constraint matrix rearranged as in the proof is

arc		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	RHS		
$m_1$	{	1	1	1																	1	
					1	1	1														1	
								1	1	1											1	
											1	1	1								1	
														1	1	1					1	
$m_2$	{	1			1			1											1	1	1	1
			1			1				1											1	
				1			1				1										1	
												1			1			1			1	
													1			1			1		1	
$m_3$	{	1										1									1	1
			1										1								2	
				1										1							2	
					1										1						2	
						1										1					2	
					1										1				2			
						1										1			2			
							1										1		2			
								1										1	2			
									1										1			
										1										1		

The corresponding graph is shown in Figure 33.



Figure 33. MCAP(3,3,2)

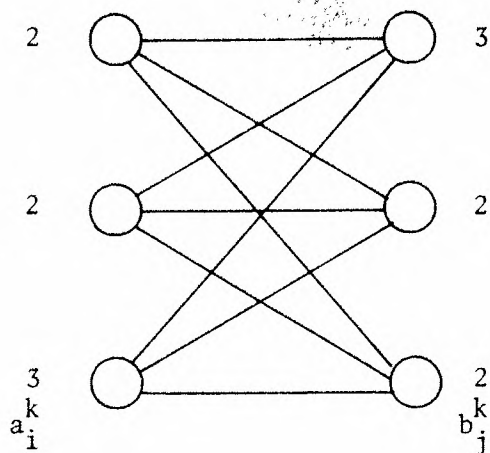
Let  $I_1 \in J_1$  be the set  $\{1,5,9,10,14,17\}$ . This is pictured in Figure 34.



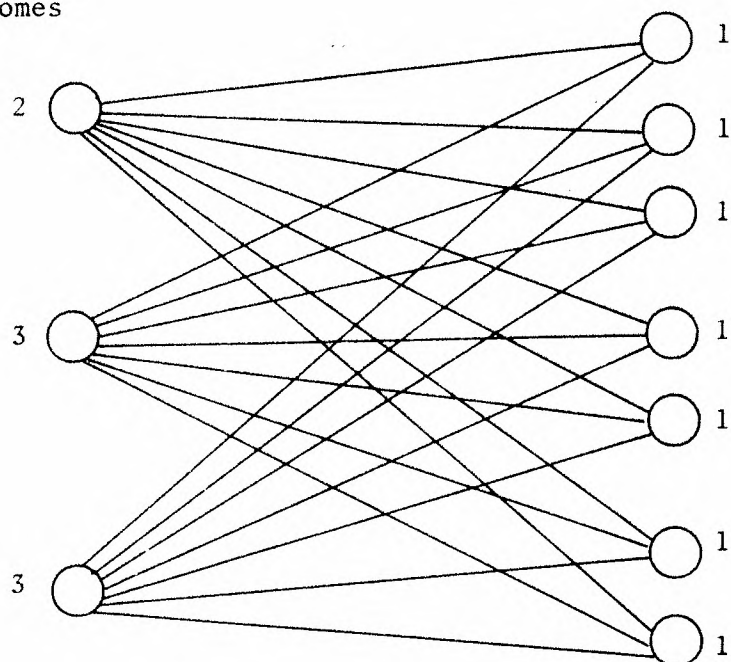
Figure 34. Independent Set in  $M_1$

Note that this set is not independent in  $M_2$  or  $M_3$  since  $\{14,17\}$  meets a single node, and  $\{1,10\}$  has more than one element.

Theorem 4.3 can be generalized to the multicommodity transportation problem quite easily. Expand each sink node with demand  $b_j^k$  into  $b_j^k$  nodes, each with demand of 1. For example, the following network



becomes



Let  $M_1$  be a partition matroid whose independent sets are all subsets of arcs, no  $a_i^k$  of which meet at the same node. For  $M_2$ , the independent sets are all subsets of arcs no two of which meet at the same node. Finally, let  $M_3$  be a partition matroid, defined similarly as for the MCAP, in which the independent sets are those subsets of arcs whose cardinality does not exceed the capacity of the original arc they represent.

Viewed in this manner, the combinatorial complexity of the MCTP is a function of the magnitude of the supplies and demands. For realistic problems this results in a phenomenal number of 0 - 1 variables.

We wish to remark that, among others, Edmonds, Karp, and Lawler have investigated matroid intersection problems for a long time with very little success, and a substantial research effort remains. We have provided yet another well-structured example of a three-matroid intersection problem. This example can provide a more convenient case to study in the search for the existence of polynomial bounded algorithms for matroid intersection problems.

## CHAPTER V

SOLUTION STRATEGIES FOR THE INTEGER MULTICOMMODITY  
TRANSPORTATION PROBLEMIntroduction

In Chapter II we showed that the MCTP(2,n,r) can be easily solved by an equivalent single commodity network problem. In this chapter we shall investigate extensions of these ideas useful in solving the general MCTP(m,n,r).

Extended Matroid Graphs

In Chapter II we saw how to construct the matroid graph for an MCTP(2,n,r), and presented a construction for a "reduced matroid graph." In this section we wish to define a natural extension of modified matroid graphs corresponding to certain relaxations of the MCTP(m,n,r) and develop an algorithm centered around this relaxed problem. Such extensions will be called extended matroid graphs (EMG).

The following construction defines an EMG.

1. Create  $mr$  source nodes for supplies  $a_i^k$ ,  $k = 1, \dots, r$ ;  $i = 1, \dots, m$
2. Create  $r(n - 1)$  sink nodes for demands  $b_j^k$ ,  $j = 1, \dots, n-1$ ;  
 $k = 1, \dots, r$
3. Create  $m$  sink nodes for capacities  $u_{in}$ ,  $i = 1, \dots, m$
4. Create a (redundant) source node for supply  $\sum_{k=1}^m u_{in} - \sum_{k=1}^r b_n^k$
5. For each variable  $x_{ij}^k$ , join an arc from the appropriate source

node to the appropriate sink node. For slacks  $s_{in}$ , join an arc from the source to the sinks created in steps 3 and 4.

We wish to make a few remarks. First, the capacity constraints for all arcs except arcs  $(i,n)$ ,  $i = 1,2,\dots,m$  are relaxed. Secondly, in step 3, we could have created the  $m$  sink nodes for capacities  $u_{ij}$  for some  $1 \leq j \leq n$  and modify steps 4 and 5 appropriately. Hence, the EMG is not unique. Finally, if  $n = 2$ , the EMG is precisely the modified matroid graph as defined in Chapter II. An example of an EMG(3,3,2) is given in Figure 35.

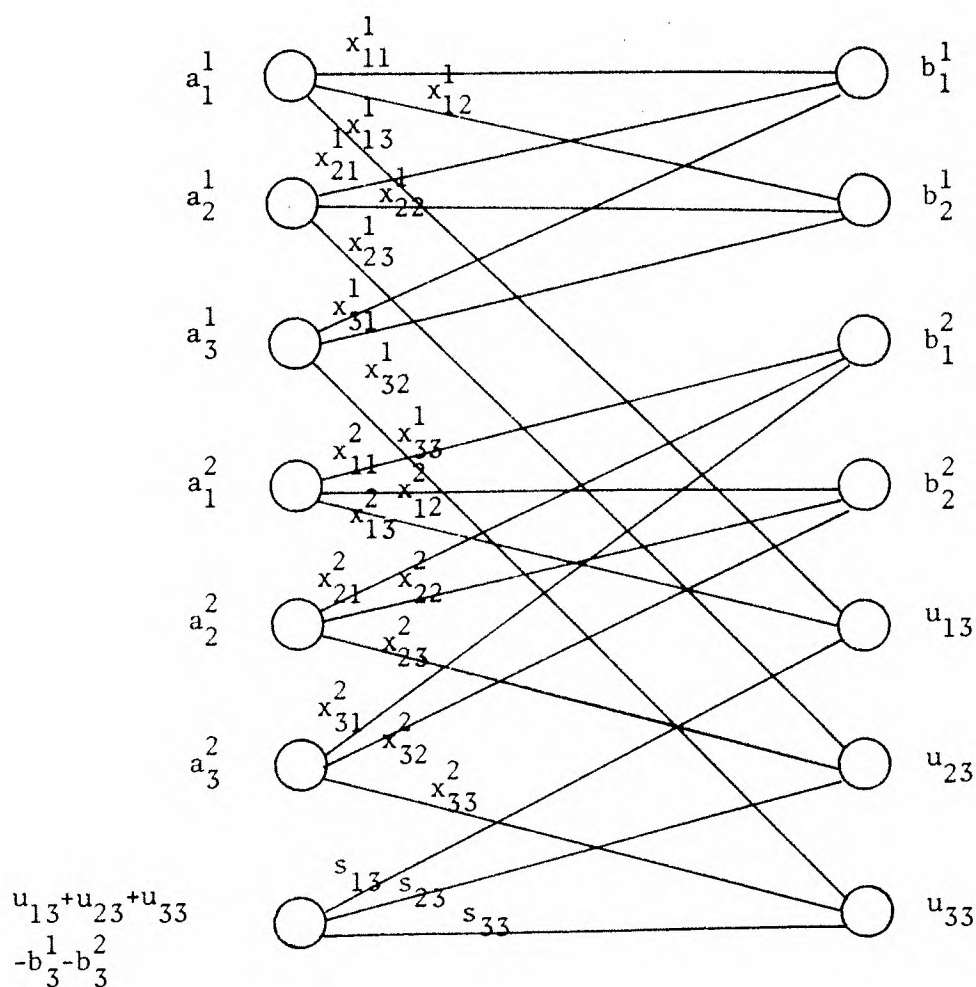


Figure 35. EMG(3,3,2)

If we solve the EMG, the relaxed capacity constraints may not be satisfied; however certain linear combinations of them are, as the following theorem illustrates.

Theorem 5.1

If there exists a feasible solution to the MCTP, then any solution to the EMG satisfies

$$\sum_{k=1}^r \sum_{j=1}^{n-1} x_{ij}^k \leq \sum_{j=1}^{n-1} u_{ij} \quad \text{for each } i$$

Proof. If a feasible solution exists, then

$$\sum_k x_{ij}^k + s_{ij} = u_{ij} \quad \text{for all } i, j$$

with  $x_{ij}^k, s_{ij} \geq 0$ .

Summing over  $j$  we have

$$\sum_j \sum_k x_{ij}^k + \sum_j s_{ij} = \sum_j u_{ij}$$

Clearly

$$\sum_k \sum_j x_{ij}^k + s_{in} \leq \sum_j u_{ij}$$

Since  $\sum_j x_{ij}^k = a_i^k$  we have



$$\sum_k a_i^k + s_{in} \leq \sum_{j=1}^{n-1} u_{ij} + u_{in}$$

$$\implies \sum_k a_i^k - (u_{in} - s_{in}) \leq \sum_{j=1}^{n-1} u_{ij}$$

$$\implies \sum_k a_i^k - \sum_k x_{in}^k \leq \sum_{j=1}^{n-1} u_{ij}$$

$$\implies \sum_k (a_i^k - x_{in}^k) \leq \sum_{j=1}^{n-1} u_{ij}$$

$$\implies \sum_k \sum_{j=1}^{n-1} x_{ij}^k \leq \sum_{j=1}^{n-1} u_{ij} \quad \text{Q.E.D.}$$

The theorem states that, for each source node, the sum of all capacity constraints on arcs incident to that source node is satisfied.

Note that if  $j = 2$ , Theorem 5.1 is precisely Theorem 2.18.

#### Corollary 5.1.1

If for some  $i$ ,  $\sum_{k=1}^r \sum_{j=1}^{n-1} x_{ij}^k > \sum_{j=1}^{n-1} u_{ij}$  then no feasible solution exists to the MCTP.

Consider the EMG(3,3,2). The relaxed constraints are

$$x_{11}^1 + x_{11}^2 \leq u_{11}$$

$$x_{12}^1 + x_{12}^2 \leq u_{12}$$

$$x_{21}^1 + x_{21}^2 \leq u_{21}$$

$$x_{22}^1 + x_{22}^2 \leq u_{22}$$

$$x_{31}^1 + x_{31}^2 \leq u_{31}$$

$$x_{32}^1 + x_{32}^2 \leq u_{32}$$

By Theorem 5.1, the following are satisfied:

$$x_{11}^1 + x_{11}^2 + x_{12}^1 + x_{12}^2 \leq u_{11} + u_{12}$$

$$x_{21}^1 + x_{21}^2 + x_{22}^1 + x_{22}^2 \leq u_{21} + u_{22}$$

$$x_{31}^1 + x_{31}^2 + x_{32}^1 + x_{32}^2 \leq u_{31} + u_{32}$$

Suppose the constraint  $x_{11}^1 + x_{11}^2 \leq u_{11}$  is violated in the optimal solution to the EMG. Since  $x_{11}^1 + x_{11}^2 - u_{11} > 0$  the constraint  $x_{12}^1 + x_{12}^2 \leq u_{12}$  must be satisfied. Therefore, for this example, at most three constraints can be violated in solving the EMG. Since  $m$  capacity constraints are explicit in the EMG, from Theorem 5.1 we have

#### Corollary 5.1.2

Solution of the EMG will result in at most  $mn - 2\max\{m,n\}$  violated capacity constraints.

The "max" operation is considered here since all constraints incident to any one node can be explicitly enforced. This is as opposed to solving the  $r$  single commodity problems independently. In this case at most  $mn - 1$  constraints may be unsatisfied.

Earlier we remarked that any sink node can be chosen in step 3 of the EMG construction. We need to show that one cannot arbitrarily choose any  $m$  capacity constraints to enforce, but only those incident to some sink node. (Note the relationship to Reban's theorem 2.16).

#### Theorem 5.2

It is impossible to construct a network satisfying all supply and

demand constraints and  $m$  capacity constraints on arcs that are not all incident to a single node.

Proof. Suppose the enforced constraints were for capacities

$u_{1j}, u_{2j}, \dots, u_{m-1,j}$  and  $u_{pq}$  where not both  $p = m$  and  $q = j$ . The arcs that must be incident to these nodes, for example, to  $u_{ij}$ ,  $1 \leq i \leq m-1$ , are  $x_{ij}^k$  and  $s_{ij}$ . Similarly for  $u_{pq}$ , the incident arcs are  $x_{pq}^k$  and  $s_{pq}$ . Assume, without loss of generality that  $\sum_j x_{ij}^k = a_i^k$  for all  $i$  and  $k$ . If  $p \neq m$ , then  $1 \leq p \leq m-1$ , but then  $x_{pq}^k$  is already incident to a node corresponding to  $u_{pq}$  if  $q = j$  or to a node corresponding to  $b_q^k$  to satisfy the demand constraints which is part of our hypothesis. This implies that the arc corresponding to  $x_{pq}^k$  would have to be duplicated. Hence  $p = m$  which leaves the only remaining value for  $q = j$  since the current network already contains the variables  $x_{mq}^k$   $q \neq j$  to satisfy the demand constraints. Q.E.D.

#### Some Apparent Theoretical Advantages of Extended Matroid Graphs

In this section we wish to discuss the applicability of EMG's in solving both the continuous and integer multicommodity transportation problem. The most valid comparison is in relation to other approaches that relax a subset of constraints, usually all the capacity constraints, leaving independent transportation problems.

The first advantage was seen in Corollary 5.1.2. A smaller number of relaxed constraints would most likely be violated in solving the EMG. Secondly, since more MCTP constraints are explicitly enforced in the EMG than in the total relaxation method, the value of the objective function after solving the EMG will be at least as high

as that obtained by solving the independent transportation problems. Hence the EMG yields a better lower bound on the objective function if one were to use a branch and bound scheme. Although this bound is generally lower than that obtained from solving the full linear program, integrality is maintained and the EMG is extremely easy to solve compared to the linear program, particularly for large problems.

Another apparent advantage is that all commodities are linked together via the EMG. The dual variables are determined in relation to all commodities. Thus for example, in a Dantzig-Wolfe approach, the master problem is smaller for the EMG, and one would expect faster convergence.

A recent article by Klingman and Russell [77] discusses a method for solving transportation problems with additional constraints. The method is basically a specialization of generalized upper bounding which Hartman and Lasdon use for the general multicommodity flow problem [44]. The "additional constraints" are the relaxed constraints of the EMG. Since these are fewer in number in the EMG formulation than in Hartman and Lasdon's approach, one would expect computational improvements.

#### A Heuristic Algorithm for the Integer MCTP

In view of the discussion in Chapter IV, large multicommodity transportation problems are exceedingly difficult to solve. A good heuristic should provide a reasonable solution quickly and will also result in a good upper bound in any branch and bound scheme. In practical applications, one would seldom resort to an optimum seeking method for the integer problem.

If the solution to the EMG is feasible to the relaxed constraints, it is clearly optimal. Otherwise we must obtain an initial feasible solution. Let  $\Gamma$  be the set of oversaturated arcs; i.e.,

$$\Gamma = \{(i,j) \mid \sum_k x_{ij}^k > u_{ij}\}$$

Notice that the EMG is an uncapacitated network problem. If we wish to enforce the capacity constraint on an arc in  $\Gamma$ , we may simply capacitate those arcs corresponding to  $x_{ij}^k$  for all  $k$  by  $x_{ij}^k \leq u_{ij}^k$  where  $\sum_k u_{ij}^k = u_{ij}$ . We have to choose the allocation of  $u_{ij}$  in order to obtain a feasible solution. This will be discussed later.

Once a feasible solution is obtained, we would like to adjust the capacities further to reduce cost. For computational purposes, the EMG was formulated as an out-of-kilter problem in which all sources are joined to a super source, and all sinks are joined to a super sink, and the super source and super sink are joined by a return arc.

The formulation of the general minimal cost circulation problem is

$$\text{minimize} \quad \sum_i \sum_j c_{ij} x_{ij}$$

$$\text{subject to} \quad \sum_j x_{ij} - \sum_i x_{ji} = 0$$

$$x_{ij} \geq l_{ij}$$

$$x_{ij} \leq d_{ij}$$

(For all EMG arcs, i.e., those which do not represent supply and demand constraints, we initially have  $\ell_{ij} = 0$  and an upper bound on arc  $x_{ij}^k$  equal to  $u_{ij}$ . Slack arcs have an upper bound of infinity.) The dual of this problem is

$$\text{maximize} \quad \sum_i \sum_j \ell_{ij} \delta_{ij} - \sum_i \sum_j d_{ij} \gamma_{ij}$$

$$\text{subject to} \quad \pi_i - \pi_j + \delta_{ij} - \gamma_{ij} \leq c_{ij}$$

$$\pi \text{ unrestricted}$$

$$\delta, \gamma \geq 0$$

The updated arc cost is defined as

$$\bar{c}_{ij} = c_{ij} + \pi_i - \pi_j$$

and  $\gamma_{ij}^* = \max\{0, -\bar{c}_{ij}\}$  is the optimal dual value. This implies that if  $\bar{c}_{ij} < 0$  then  $\gamma_{ij}^* > 0$  and increasing  $d_{ij}$  will improve the value of the objective function.

Assume that a feasible solution to the EMG has been found with  $\Gamma$  being the set of capacitated arcs. If  $u_{ij}^k = u_{ij}$  we cannot further increase  $u_{ij}^k$  and if  $u_{ij}^k = 0$  we cannot decrease it. Table 1 shows the change in the value of the objective function for increasing and decreasing  $u_{ij}^k$  given a particular capacitated state.

A potential improvement exists only if  $\bar{c}_{ij}^k < 0$  and  $u_{ij}^k < u_{ij}$ . We emphasize "potential" since a capacity change may force some arc  $(i, j) \notin \Gamma$  to be violated, or result in a basis change if we exchange capacity or a pair of arcs for which  $\bar{c}_{ij}^k - \bar{c}_{ij}^l < 0$ . This may result

Table 1. Sensitivity Analysis for Arc Capacities

		$u_{ij}^k = 0$	$0 < u_{ij}^k < u_{ij}$	$u_{ij}^k = u_{ij}$
Increase $u_{ij}^k$	$\bar{c}_{ij}^k < 0$	$\bar{c}_{ij}^k$	$\bar{c}_{ij}^k$	
	$\bar{c}_{ij}^k \geq 0$	0	0	
Decrease $u_{ij}^k$	$\bar{c}_{ij}^k < 0$		$-\bar{c}_{ij}^k$	$-\bar{c}_{ij}^k$
	$\bar{c}_{ij}^k \geq 0$		0	0

in a higher cost since the dual values are meaningless if the basis changes. From these considerations, we define a local optimum as a solution such that no capacity exchange that preserves feasibility and results in a lower cost can be made. We immediately see the following:

### Theorem 5.3

If for all  $\bar{c}_{ij}^k < 0$  we have  $u_{ij}^k = u_{ij}$  then the current solution is globally optimal.

We will now describe an algorithm that will produce at least a local optimum. The general idea is to choose the most negative  $\bar{c}_{ij}^k$ , determine if a potential improvement exists, and then to determine the maximum increase in  $u_{ij}^k$ .

Assume that  $\bar{c}_{ij}^k$  is the most negative. First set  $u_{ij}^k = u_{ij}^k + 1$ . The arc corresponding to  $x_{ij}^k$  is now out-of-kilter. This problem is now resolved and if the network has a feasible solution, a breakthrough path is found. Notice that by increasing  $u_{ij}^k$  by unity we have violated the MCTP constraint on arc  $(i,j)$ . As long as the same basis is



maintained, the same breakthrough path will be found if  $u_{ij}^k$  is further increased. The next step is to determine how much  $u_{ij}^k$  can be increased and maintain feasibility with respect to the capacity constraints of the MCTP. The blocking conditions may be any one of the following:

- (i) No MCTP capacity constraint may be violated.
- (ii) All flows must be non-negative.
- (iii)  $\sum_k u_{ij}^k = u_{ij}$ .

Let  $\Phi$  be the cycle formed by the out-of-kilter arc  $x_{ij}^k$  and the breakthrough path, oriented in the direction of the arc corresponding to  $x_{ij}^k$ . We now determine the blocking conditions. Let

$$\Delta_1 = u_{ij} - u_{ij}^k$$

$\Delta_1$  is the maximum allowable increase of  $u_{ij}^k$ . Let

$$\Delta_2 = \min\{x_{pq}^s \text{ for all reverse arcs in } \Phi\}$$

$\Delta_2$  determines the maximum allowable increase before any basic variable becomes negative. Notice that the flows on all arcs not in  $\Phi$  will not change. Let

$$\Delta_3 = \min\{u_{pq}^s - x_{pq}^s \text{ for all forward arcs in } \Phi\}$$

We are only considering reallocation of capacity of arc  $(i,j)$ , hence  $\Delta_3$  represents the maximum increase in flow before any variable reaches its current upper bound.

If  $(p,q) \notin \Gamma$ , then  $u_{pq}^s = u_{pq}$ . For any arc  $(p,q) \in \Gamma$ , let



$\theta_{pq}^s = 1$  if  $x_{pq}^s$  is a forward arc in  $\Phi$ , and  $\theta_{pq}^s = -1$  if  $x_{pq}^s$  is a reverse arc in  $\Phi$ . Let

$$\Delta_4 = \left[ \frac{(u_{pq} - \sum_s x_{pq}^s)}{\sum_s \theta_{pq}^s} \right] \quad \text{if} \quad \sum_s \theta_{pq}^s > 0$$

$$= \infty \text{ otherwise}$$

where  $[ \ ]$  denotes the greatest integer function. The sum over  $s$  of  $\theta_{pq}^s$  represents the net change in flow on arc  $(p,q)$  per unit increase of flow around the cycle. If this sum is non-positive then no violation of capacity can occur; otherwise  $\Delta_4$  represents the net increase of flow on arc  $(p,q)$  per unit change in  $x_{ij}^k$ . This establishes blocking condition (i). Finally, let

$$\Delta_5 = \sum_{\ell} u_{ij}^{\ell} \quad \text{for } x_{ij}^{\ell} \text{ reverse arc in } \Phi$$

$$+ \sum_{\ell} (u_{ij}^{\ell} - x_{ij}^{\ell}) \quad \text{for } x_{ij}^{\ell} \notin \Phi$$

$\Delta_5$  makes sure that we can preserve condition (iii) by decreasing the capacity of other commodities for arc  $(i,j)$ . Let

$$\Delta = \min\{\Delta_1, \Delta_2, \Delta_3, \Delta_4, \Delta_5\}.$$

If  $\Delta = 0$ , choose the next minimum  $\bar{c}_{ij} < 0$  and repeat. If no such  $\bar{c}_{ij}$  exists, terminate. If  $\Delta > 0$ , set  $u_{ij}^k = u_{ij}^k + \Delta$  and decrease the capacities on other commodities corresponding to arc  $(i,j)$  by a total of  $\Delta$  units. The EMG is resolved and the process is continued until

a local optimum is attained.

The algorithm was coded in FORTRAN IV and several test problems were run to evaluate performance of the heuristic. Computational experience is presented in Table 2. The important result is the comparison between the unconstrained solution in which all capacity constraints are relaxed and the initial EMG solution. In many cases, the number of oversaturated arcs are reduced in the EMG, and in nearly all cases, the total infeasibility was reduced and the lower bound was higher. The cases in which the unconstrained solution and EMG solution were equal merely indicates that the capacity constraints enforced in the EMG were not binding and redundant. This aspect will be discussed in the concluding chapter. The column labeled P is the percentage in cost that the local optimum obtained is away from the lower bound, determined by the initial EMG solution. In more than 50 percent of the cases this percentage was less than 4 percent and in all but one case was less than 13 percent. Since the EMG solution is infeasible and provides a lower bound, the deviation from the true optimum is actually less.

In a few cases, a feasible solution could not be obtained by the technique used here. This has no effect on the performance of the heuristic itself, however. There are countless ways of obtaining an initial feasible solution, and a complete computational study may determine an improved method. We have used the following heuristic. Let  $(p,q) \in \Gamma$ . Let  $k_1, k_2, \dots, k_r$  be such that

$$x_{pq}^{k_1} \geq x_{pq}^{k_2} \geq \dots \geq x_{pq}^{k_r}$$

Table 2. Computational Results

Problem Characteristics					Unconstrained Solution			EMG Solution					
m	n	r	TS	CAP	NA	TI	COST	C <sub>1</sub>	NA	TI	C <sub>2</sub>	P	TIME
3	3	2	915	120	2	110	78905	78905	2	110	81820	3.7	.165
3	3	2	402	50	3	146	65024	65688	3	84	73074	11.2	.351
3	3	2	885	120	2	162	98677	137672	1	57	138698	0.7	.133
3	3	2	966	130	3	390	195884	202518	3	124	208966	3.2	.187
3	3	2	657	88	3	162	138168	140868	2	86	144566	2.6	.182
3	3	2	276	40	3	62	39126	39536	2	42	44408	12.3	.190
											Avg.	5.6	.201
3	3	5	2616	600	0*	0*	470972*	470972*	0*	0*	--	--	.433
3	3	5	4110	550	2	627	804904	804904	2	627	891401	10.7	.733
3	3	5	1083	150	3	449	188889	193953	2	174	206463	6.5	.799
3	3	5	2316	300	2	109	165802	166264	1	87	168178	1.2	.327
3	3	5	2775	340	4	512	--	**	--	--	--	--	--
											Avg.	6.1	.573
5	5	2	1375	75	4	110	181879	185645	1	44	188505	1.5	1.092
5	5	2	340	30	9	231	62831	69328	5	124	88851	28.2	1.096
5	5	2	1635	120	1	69	226053	226053	1	69	226812	0.3	.517
5	5	2	1070	100	2	98	121612	121766	1	70	122536	0.6	.480
											Avg.	7.65	.796
5	5	5	1860	180	2	60	242327	242327	2	60	246167	1.6	1.848
5	5	5	2615	250	3	181	303047	303047	3	181	315029	4.0	2.310
5	5	5	4055	400	3	232	404260	**	--	--	--	--	--
5	5	5	4775	450	5	477	372737	373604	4	375	399284	6.87	2.376
5	5	5	7160	745	2	90	843423	843423	2	90	848163	0.6	2.248
											Avg.	3.27	2.196

\*Solution was optimal.

\*\*No initial feasible solution obtained.

LEGEND: TS = Total supply =  $\sum_i \sum_k a_i^k$

CAP = Arc capacities

NA = Number of oversaturated arcs

TI = Total infeasibility

C<sub>1</sub> = Unconstrained EMG solution costC<sub>2</sub> = Local optimum costP = 100 (C<sub>2</sub> - C<sub>1</sub>)/C<sub>1</sub>

TIME = Execution time exclusive of I/O, in seconds of CPU on U1108.

1.  $s = 0, i = 0$

2.  $i = i + 1$

3. If  $x_{pq}^k \leq u_{pq} - s$ , set  $u_{pq}^k = x_{pq}^k$  and  $s = s + u_{pq}^k$  and return to step 2. Otherwise set  $u_{pq}^k = u_{pq} - s$  and  $s = u_{pq}^k$ . Then for  $j > i$  set  $u_{pq}^j = 0$ .

This procedure is intuitively reasonable. We would like a feasible solution to be "close" to the unconstrained solution, so we choose to allocate the highest capacity to arcs with the greatest flow.

Based on this initial allocation, the EMG is resolved as a semi-capacitated network problem. If it is not feasible, we reallocate the capacity among the commodities of arcs in  $\Gamma$  as follows. For the first arc in the set  $\Gamma$ , say  $(p,q)$ , let

$$\Delta = [\alpha u_{pq}] + 1$$

where brackets denotes the greatest integer function and  $0 < \alpha < 1$ ,

say  $\alpha = .10$ . Then set  $u_{pq}^r = u_{pq}^r - \Delta$ ,  $u_{pq}^{r-1} = u_{pq}^{r-1} + \Delta$ ,  $u_{pq}^{r-2} = u_{pq}^{r-2} - \Delta$ , etc. for an even number of commodities. Any variation of

this procedure could also be used. The EMG is resolved and if it is still infeasible, another arc from  $\Gamma$  is chosen and the above procedure is repeated.

## CHAPTER VI

SOME TECHNIQUES APPLICABLE TO THE GENERAL INTEGER  
MULTICOMMODITY FLOW PROBLEM

In this chapter we wish to discuss some methods that are not based on extended matroid graphs and hence not restricted to the MCTP but applicable to general multicommodity network flow problems.

The General Integer Cycle Formulation

In Chapter III we observed that a basis to a multicommodity flow problem consists of a set of rooted spanning forests and a set of basic cycles. In fact, any feasible solution can be expressed as a set of rooted spanning forests and a set of non-negative flows on cycles formed by the out-of-tree arcs. This fact was first observed and utilized by Saigal [113] and later by Hartman and Lasdon [44] in their respective algorithms. Conceptually, then, one may view the integer MCTP as

- (i) choosing any set of rooted spanning forests for the individual commodities, and
- (ii) finding integer flows on the cycles determined by the out-of-tree arcs that satisfy the capacity constraints at minimal cost.

Note that conservation of flow constraints are automatically satisfied by nature of the cycles. From this viewpoint, we need only work explicitly with the cycle matrix generated by the set of out-of-tree arcs for any given set of rooted spanning forests.

We now wish to mathematically characterize the cycle problem. For commodity  $k$ , and a network consisting of  $m$  arcs, let  $C^k$  be an  $m \times p$  matrix in which each of the  $p$  columns represents the vector expression of the cycles formed by the out-of-tree arcs with respect to the forest  $F^k$ , where the coefficient of cycle  $C_j^k$  is +1 if arc  $j$  is an out-of-tree arc (in other words, we orient the cycle in the direction of the out-of-tree arc). Cycle  $C_j^k$  is generated by the flow variable  $x_j^k$ . We may easily compute the updated cost coefficient  $\bar{c}_j^k$  by a simple labelling procedure on the network as is done in Johnson's primal algorithm [70]. The integer cycle problem is then

$$\text{CP:} \quad \text{minimize} \quad \sum_{k=1}^r \sum_{j \in C^k} \bar{c}_j^k \theta_j^k$$

subject to

$$[C^1 \mid C^2 \mid \dots \mid C^r] \begin{bmatrix} \theta^1 \\ \theta^2 \\ \vdots \\ \theta^r \end{bmatrix} \leq \underline{u} - \sum_k \underline{x}^k \quad (6.1)$$

$$-C^k \underline{\theta}^k \leq \underline{x}^k \quad \text{for all } k \quad (6.2)$$

$$\underline{\theta}^k \geq 0 \text{ and integer for all } k \quad (6.3)$$

where  $\underline{x}^k$  is the current solution vector generated by  $F^k$ .

Constraints (6.1) guarantee that the total net increase in flow on any arc cannot exceed the residual capacity; (6.2) states that, for each commodity the total net reduction in flow on any arc for any

commodity cannot exceed its current flow (implying non-negativity), and (6.3) guarantees non-negativity on the out-of-tree arcs. Actually (6.3), except for the integer restriction, is explicitly incorporated in (6.2) and therefore is redundant. To illustrate consider the example in Figure 22 and the forests given in Figure 23. The cycle matrices are (for convenience, we have numbered the arcs in such a way that arc (1,1) is arc 1, arc (1,2) is arc 2, ..., arc (3,3) is arc 9).

arc	$C_1^1$	$C_2^1$	$C_4^1$	$C_8^1$	arc	$C_3^2$	$C_5^2$	$C_7^2$	$C_9^2$
1	1				1	-1	1	-1	-1
2		1			2		-1	1	1
3	-1	-1			3	1			
4			1		4	1	-1		1
$C^1 = 5$		-1		-1	$C^2 = 5$		1		
6		1	-1	1	6	-1			-1
7	-1		-1		7			1	
8				1	8			-1	-1
9	1		1	-1	9				1

$$\underline{x}^1 = (0, 0, 2, 0, 2, 0, 2, 0, 0)$$

$$\underline{x}^2 = (2, 0, 0, 0, 0, 2, 0, 2, 0)$$

We wish to point out that if integrality is relaxed, the above problem will solve the continuous multicommodity flow problem. (In this example, the optimal linear programming solution to the cycle problem is  $\theta_1^1 = 3/2$  and  $\theta_9^2 = 3/2$ ). We also wish to point out that the continuous cycle problem is equivalent to Hartman and Lasdon's generalized upper bounding procedure. For if, at any iteration, the pivot



element occurs in (6.2), then this simply implies a change in a rooted spanning forest; if pivoting occurs in (6.1) then an arc becomes saturated. This implies that had we chosen initially the forests that are optimal for the LP, then we may solve the problem only by considering constraints (6.1).

The cycle formulation can be utilized in several ways; these will be explored in subsequent sections.

### A Combined Relaxation-Cutting Plane Strategy

Let us assume that we have solved the optimal linear program. We have obtained a set of forests  $\{F^k\}$  and a set of basic cycles, assumed to be non-integer. One method of solving the integer problem is to apply a cutting plane to the updated linear programming tableau of CP. We will show that it is necessary only to explicitly consider a small subset of rows from CP. Define the cycle tableau as a matrix consisting only of the rows of (6.1) corresponding to saturated arcs in the optimal tableau to CP.

It is not necessary to explicitly solve CP to obtain the cycle tableau. If the MCTP were solved by the Hartman and Lasdon algorithm, the optimal solution would yield  $S_1^{-1}$  (defined in Chapter III) the inverse of the cycle matrix restricted to saturated arcs (the "working basis"). The cycle tableau can be generated graph-theoretically. Any column in the updated cycle tableau is of the form  $y_j^k = S_1^{-1} a_j^k$  where  $a_j^k$  is the vector expression of a cycle restricted to the saturated arcs. Since each  $a_j^k$  contains only +1, -1, or 0 components,  $y_j^k$  is simply a  $\pm 1, 0$  linear combination of columns of  $S_1^{-1}$ . But each column of  $S_1^{-1}$  corresponds to a saturated arc, and each column of the cycle



tableau by definition corresponds to a cycle. This suggests the following method for generating any updated column  $y_j^k$  in the cycle tableau.

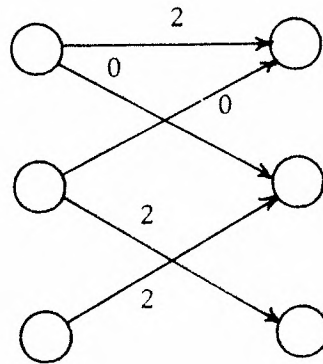
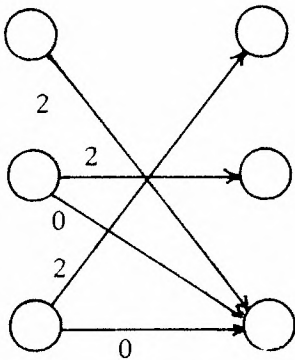
Step 0: Initialize  $y_j^k$  to zero.

Step 1: Determine the cycle formed by  $x_j^k$ . Orient the cycle in the direction of  $x_j^k$ .

Step 2: Trace through the cycle. If an arc in the cycle, say  $x_\ell^k$  is saturated and traversed in the forward direction, then  $y_f^k = y_j^k + (S_1^{-1})_\ell$ ; if it is traversed in the reverse direction, then  $y_j^k = y_j^k - (S_1^{-1})_\ell$ .

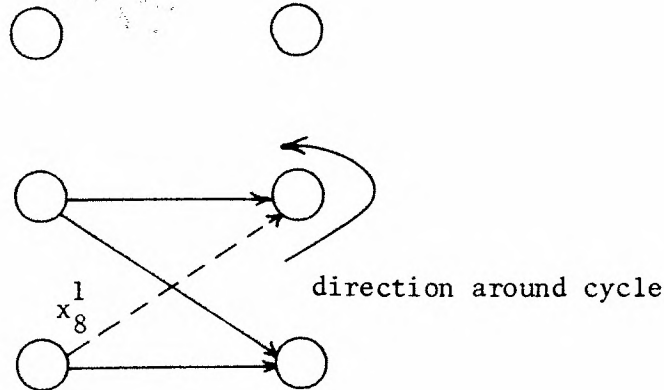
Step 3: Repeat Step 2 until all arcs in the cycle are traversed.

To illustrate, consider the example presented in Chapter II. The optimal linear programming bases are



$$\text{with } S_1^{-1} = \begin{bmatrix} 1/2 & 1/2 \\ -1/2 & 1/2 \end{bmatrix}$$

corresponding to saturated arcs 1 and 9. Consider  $y_8^1$ . Arc  $x_8^1$  forms the following cycle



Since the cycle crosses arc 9 in the reverse direction, then

$$y_8^1 = - \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}.$$

The complete cycle tableau is (note that the updated cost coefficients are easily determined):

Basis	$x_1^1$	$x_2^1$	$x_4^1$	$x_8^1$	$x_3^2$	$x_5^2$	$x_7^2$	$x_9^2$	$s_1$	$s_9$	RHS
	0	2	11	4	2	5	9	0	8	10	
$x_1^1$	1	0	1/2	-1/2	-1/2	1/2	-1/2	0	1/2	1/2	3/2
$x_9^2$	0	1	1/2	-1/2	1/2	-1/2	1/2	1	-1/2	1/2	3/2

From Theorem 3.5 we require that the values of all cycles be integer. Since all other capacity constraints are non-binding, they need not be considered. By only considering the cycle tableau we have also relaxed the non-negativity restrictions, (5.2). Consider a Gomory cut from the cycle tableau. For the above example we have

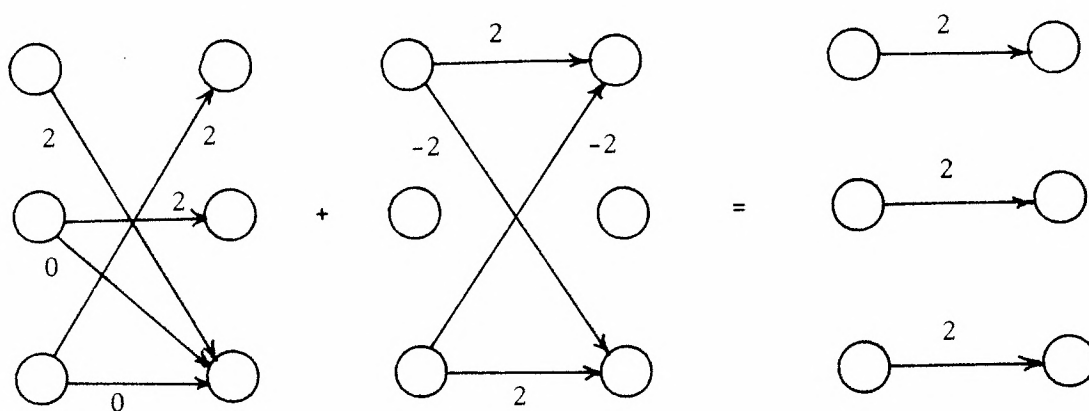
$$-\frac{1}{2}x_4^1 - \frac{1}{2}x_8^1 - \frac{1}{2}x_3^2 - \frac{1}{2}x_5^2 - \frac{1}{2}x_7^2 - \frac{1}{2}s_1 - \frac{1}{2}s_9 \leq -\frac{1}{2}$$

The dual simplex criterion specifies  $x_3^2$  to enter in the cut row. The

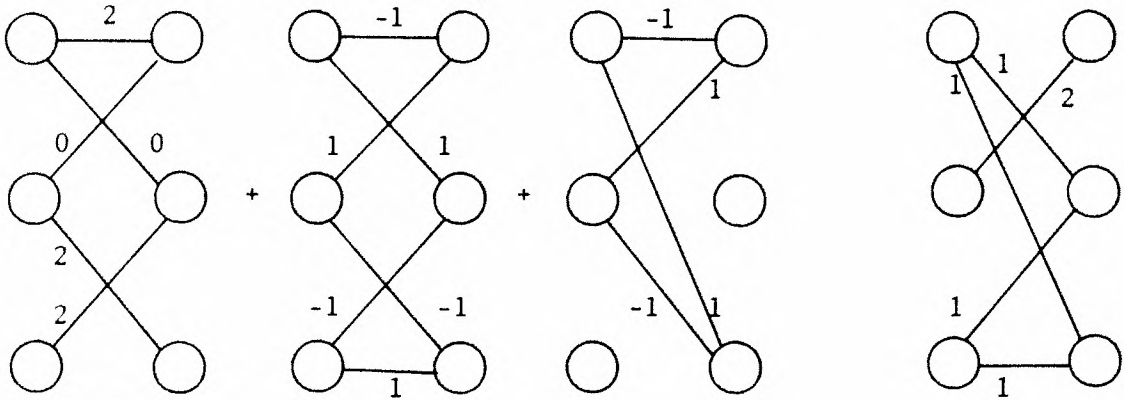
pivot yields the following tableaux:

Basis	$x_1^1$	$x_2^1$	$x_4^1$	$x_8^1$	$x_3^2$	$x_5^2$	$x_7^2$	$x_9^2$	$s_1$	$s_9$	$t_1$	RHS
	0	2	11	4	2	5	9	0	8	10	0	
$x_1^1$	1	0	1/2	-1/2	-1/2	1/2	-1/2	0	1/2	1/2	0	3/2
$x_9^2$	0	1	1/2	-1/2	1/2	-1/2	1/2	1	-1/2	1/2	0	3/2
$t_1$	0	0	-1/2	-1/2	-1/2	-1/2	-1/2	0	-1/2	-1/2	1	-1/2
	0	2	9	2	0	3	7	0	6	8	4	
$x_1^1$	1	0	1	0	0	1	0	0	1	1	-1	2
$x_9^2$	0	1	0	-1	0	-1	0	1	-1	0	1	1
$x_3^2$	0	0	1	1	1	1	1	0	1	1	-2	1

The optimal integer solution is



Commodity 1



Commodity 2

In this example we encountered no difficulties with non-negativity restrictions. Suppose that  $\bar{c}_4^1 = 1$ . Then the dual simplex criterion for the cut row specifies that  $x_4^1$  becomes basic. However  $x_6^1$  becomes negative as can be seen graph-theoretically from the cycle generated by  $x_4^1$ . We can now enforce the non-negativity constraint on  $x_6^1$ . This is

$$-x_2^1 + x_4^1 - x_8^1 \leq 0$$

The tableau is then

Basis	$x_1^1$	$x_2^1$	$x_4^1$	$x_8^1$	$x_3^2$	$x_5^2$	$x_7^2$	$x_9^2$	$s_1$	$s_9$	$s_6^1$	$t_1$	RHS
	0	2	1	4	2	5	9	0	8	10	0	0	
$x_1^1$	1	0	1/2	-1/2	-1/2	1/2	-1/2	0	1/2	1/2	0	0	3/2
$x_9^2$	0	1	1/2	-1/2	1/2	-1/2	1/2	1	-1/2	1/2	0	0	3/2
$s_6^1$	0	-1	1	-1	0	0	0	0	0	0	1	0	0
$t_1$	0	0	-1/2	-1/2	-1/2	-1/2	-1/2	0	-1/2	-1/2	0	1	-1/2

	0	2	0	3	1	4	8	0	7	9	0	2	
$x_1^1$	1	0	0	-1	-1	0	-1	0	0	0	0	1	1
$x_9^2$	0	1	0	-1	0	-1	0	1	-1	0	0	1	1
$s_6^1$	0	-1	0	-2	-1	-1	-1	0	-1	-1	0	2	-1
$x_4^1$	0	0	1	1	1	1	1	0	1	1	0	-2	1

We may now employ the dual simplex on the violated constraint yielding

Basis	$x_1^1$	$x_2^1$	$x_4^1$	$x_8^1$	$x_3^2$	$x_5^2$	$x_7^2$	$x_9^2$	$s_1$	$s_9$	$s_6^1$	$t_1$	RHS
	0	1	0	1	0	3	7	0	6	8	0	4	
$x_1^1$	1	1	0	1	0	1	0	0	1	1	0	-1	2
$x_9^2$	0	1	0	-1	0	-1	0	1	-1	0	0	1	1
$x_3^2$	0	1	0	2	1	1	1	0	1	1	0	-2	1
$x_4^1$	0	-1	1	-1	0	0	0	0	0	0	0	0	0

The integer solution happens to be the same as in the original example.

We may summarize these observations with the following algorithm.

Algorithm RCP:

Step 0: Solve the linear program obtaining the optimal forests and  $S_1^{-1}$ .

Step 1: If all basic cycles are integer, terminate. Otherwise generate the cycle tableau as described.

Step 2: Choose the topmost row for which the right hand side is fractional. Generate a Gomory cut and pivot.

Step 3: Determine if the resultant flow values yield a feasible solution. If not, add the appropriate non-negativity

constraints (or capacity constraints) and apply the dual simplex criterion until feasibility is attained. If all cycles are integer-valued, terminate; otherwise repeat Step 2.

Since Algorithm RCP is simply Gomory's cutting plane algorithm applied in a special manner to take advantage of the network structure, finiteness readily follows. The advantage to this method as opposed to a usual direct application of cutting planes is that only a very small number of rows need be considered in the tableau (empirical evidence [39] has shown that in the optimal LP solution to multicommodity flow problems, only a very small percentage of arcs are saturated). The non-binding constraints are superfluous unless a cut pivot violates one of them.

#### Other Solution Strategies

Due to the large size of multicommodity flow problems, it is doubtful that any good procedure that yields a global optimum can be developed. In this section we shall suggest heuristic procedures that can apply to the general problem.

One such procedure that immediately comes to mind is a procedure similar to the heuristic described in Chapter V for the MCTP. For the general problem, however, one cannot link the commodities together in one network as was done for the MCTP using extended matroid graphs. The problems must be solved independently, and consequently the dual variables do not provide as much information than if they were computed relative to all commodities. One could, however, utilize the

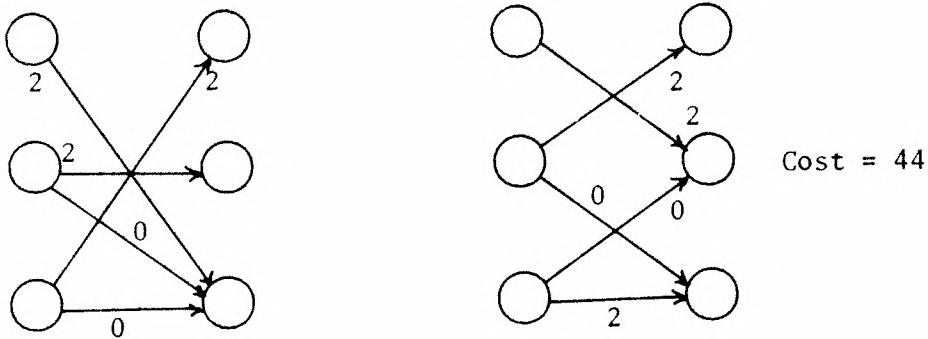
dual variables on saturated arcs to heuristically allocate capacity among the individual commodities.

We will next consider neighborhood search algorithms.

Define (cf. [34], p. 325) the unit neighborhood of a point  $x^*$  as

$$R(x^*) = \{x \mid x_j = x_j^* - 1, x_j^*, x_j^* + 1, \forall j\}$$

Also define the  $m$ -variable neighborhood  $N_m(x^*)$  as the set of integer vectors, each of which differs from  $x^*$  in not more than  $m$  components. It is not computationally practical to work with  $R(x^*)$  as we shall shortly see. Consider  $N_1(x^*)$ , the 1-variable neighborhood. Given the following solution to the example considered throughout this chapter

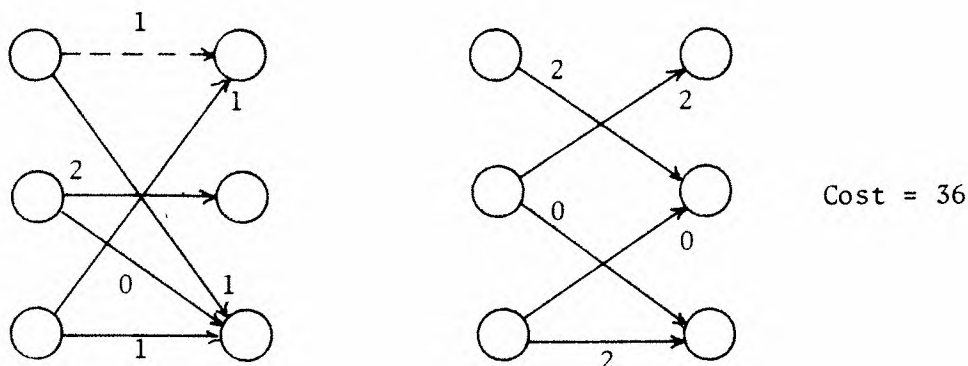


we have that

$$\begin{aligned} \bar{c}_1^1 &= -8 \\ \bar{c}_2^1 &= 2 \\ \bar{c}_4^1 &= 6 \\ \bar{c}_8^1 &= 9 \end{aligned}$$

$$\begin{aligned} \bar{c}_1^2 &= 2 \\ \bar{c}_3^2 &= 7 \\ \bar{c}_5^2 &= 0 \\ \bar{c}_7^2 &= 14 \end{aligned}$$

We wish to increase  $x_1^1$ , but must stop at 1 since arc 9 becomes saturated.



No adjacent solution (in the 1-variable neighborhood) is better but the solution is not optimal. We have reached a local optimum using  $N_1$ . Had we used the 2-variable neighborhood, we would find that by simultaneously increasing the flow on  $x_3^2$  and  $x_1^1$  by 1 from the current solution we would reach the optimum. However, using a 2-variable neighborhood we have greatly increased the computational complexity of the problem since all simultaneous changes of two variables must be considered. One can easily see that the complete unit neighborhood would be extremely difficult to use since  $|R(x^*)| = 3^n$  where  $n$  is the number of variables.

Small variable neighborhoods can be used for approximate solutions to large MCTP's. From this point of view we have several strategies for choosing the initial forests,

- (i) We may choose dual feasible bases, i.e., the unconstrained solutions to the individual commodities. Here all  $\bar{c}_j^k \geq 0$ .
- (ii) We may select a primal feasible basis. This may not be trivial because of the capacity constraints.



- (iii) We may choose any feasible integer solution (not necessarily a basis). This is possible by the following.

Lemma 6.1 Let  $\underline{x}$  be a feasible integer solution to a MCTP. Then  $\underline{x}$  can be decomposed into a set  $F$  of rooted spanning forests and a set  $C$  of cycles, fundamental to members of  $R$ .

Proof. The proof is by construction. Let  $X^k = \{x_j^k \mid x_j^k > 0\}$  that is, the set of flow variables for commodity  $k$  which are strictly positive. From  $X^k$ , choose elements  $x_j^k$  one at a time so that no cycle is formed. If  $X^k$  is exhausted and a rooted spanning forest is not formed add the appropriate arcs  $x_j^k = 0$  to obtain a degenerate basis. Do the same if  $X^k$  is not exhausted but addition of any other arc from the current list forms a cycle. If now  $X^k = \emptyset$ , terminate. If  $X^k \neq \emptyset$ , addition of any other arc forms a cycle. Continue until  $X^k = \emptyset$  for all  $k$ . The construction is complete.

The latter two starting points are appropriate if one were to employ a heuristic method and were only concerned with approximations. In these cases we would also have to consider decreasing the flow on a cycle if the flow is positive.

## CHAPTER VII

### CONCLUSIONS AND RECOMMENDATIONS

The primary objective of this research was to study the nature of integer solutions in multicommodity network flow problems. The principal results can be summarized as follows.

Some new and rather unique results concerning the multicommodity transportation problem were established; namely, a necessary and sufficient condition for a totally unimodular constraint matrix, and an equivalent single commodity network flow problem for that class of problems.

The theoretical development relating graphic matroids and linear programming was initiated. This was motivated by the question of characterizing linear programs that can be solved by equivalent network problems.

The nature of cycles in directed multicommodity networks were considered in relation to the non-unimodular aspects of the problem. A graph-theoretic condition was developed whereby one can determine if integral solutions can be obtained.

Matroid theory was again applied to show that apparently no efficient algorithms exist for the integer multicommodity transportation problem.

A single commodity network for the general multicommodity transportation problem with certain relaxed constraints was proposed. A heuristic algorithm was constructed around this network and some

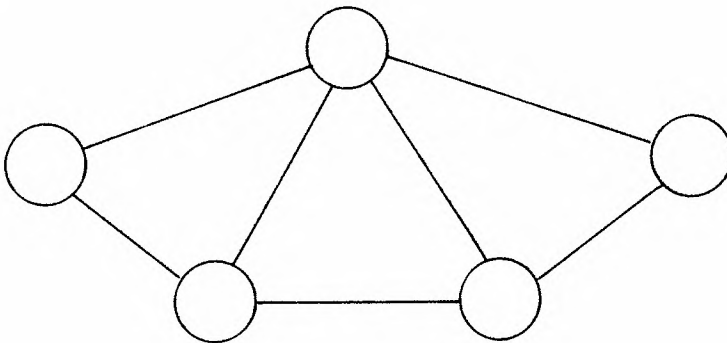
computational results were reported. Finally, some approaches to the general integer multicommodity flow problem were discussed.

The relationship between graphic matroids and linear programming appears to have opened a new area of research in mathematical programming and network flows. Many aspects of this general problem and the integer multicommodity flow problem should be considered in future research endeavors.

We wish to propose a conjecture concerning the general question of when a constraint matrix of 0's and 1's can be transformed into an equivalent network problem. Two examples were given in Chapter II, one in which the transformation was possible, and one in which it was not. As a third example, let  $A$  be the following:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

The graph of  $A$  is



We obtain a directed graph by multiplying rows 2 and 4 by  $-1$ .

Let us examine these three cases. In the first example,  $G$  is bipartite; in the last two,  $G$  is not. However, in the last case, we may delete a node (corresponding to a linearly dependent constraint) such that the remaining graph has no odd cycles.

The general question that we are considering is under what conditions can a linear program with a zero-one coefficient matrix be solved as a pure network programming problem. The answer appears to be associated with odd cycles in the matroid graph. In the second example,  $N$  is not totally unimodular; however, the unimodular property is not preserved under mod 2 row operations. To illustrate this, let

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

$B$  can be derived from  $A$  through mod 2 row operations;  $A$  is totally unimodular,  $B$  however, is not.

From these observations, we propose the following conjecture. A proof or counterexample could not be found.

Conjecture. Let  $A$  be a matrix of 0's and 1's whose corresponding matroid is graphic with matroid graph  $G$ . Then the linear program

$$\begin{aligned} \min \quad & \underline{c} \underline{x} \\ \text{subject to} \quad & \underline{Ax} = \underline{b} \\ & \underline{x} \geq \underline{0} \end{aligned}$$

can be solved as a pure network programming problem if and only if deletion of any one vertex of  $G$  results in a graph with no odd cycles.

If this conjecture is true, the following theorem would be useful.

### Theorem

Let  $R = [I, N]$  be a standard representative matrix of a graphic matroid. If every column of  $N$  contains an odd number of ones, then  $G$  is bipartite.

Proof. Each column  $j$  of  $N$  together with the rows of  $R$  corresponding to the ones in column  $j$  define a cycle of  $G$ . Clearly every cycle defined by columns of  $N$  is even. It is well known that the cycles corresponding to columns of  $N$  form a basis for the cycle subspace of  $G$  (Seshu and Reed [116], and Liu [85]), and that the set of cycles in a graph form an Abelian group under  $\oplus$ , where  $\oplus$  is a set operation defined by  $X \oplus Y = (X \cup Y) - (X \cap Y)$ . Let  $c_1$  and  $c_2$  be the edge sets of any two cycles of  $N$ . If

$$(c_1 \cup c_2) - (c_1 \cap c_2) = \emptyset$$

the result is an edge disjoint union of cycles which is even. If  $c_1 \cap c_2$  contain an odd number of edges and hence  $c_1 \oplus c_2$  is even since the sum of two odd numbers is even. If  $c_1 \cap c_2$  is even, then clearly  $c_1 \oplus c_2$  is even. By applying the argument to all cycles generated by columns of  $N$ , it follows that every cycle in  $G$  is even and hence  $G$  is bipartite. Q.E.D.

Klingman and Russell [77] point out that several constrained single commodity network problems have equivalent pure network

representations. Settlement of the conjecture would aid in developing a unified theory of these types of problems. One should also attempt to identify other classes of integer programs that meet the graphic matroid requirement.

All feasible integer solutions to the MCAP are vertices of the linear programming polytope. The MCTP has many integer vertices; in fact, only a small percentage of basis columns could possibly destroy integrality, and this has been partially characterized. One should investigate "how bad" the MCTP polytope is with regard to the density of non-integer vertices. This, and the results of Chapter III may lead to an algorithm for determining a good, or possibly the best, integer vertex, and provide a good approximation for the MCTP.

We have seen that the capacitated arcs cause all the problems of non-unimodularity in multicommodity networks. Corollary 3.7.1 gives a sufficient condition for integer solutions. Some interesting questions that arise are the following. First, can one identify other general classes of network structures with all capacitated arcs that satisfy Corollary 3.7.1 like the  $MCTP(2,n,r)$ ? Secondly, if a multicommodity network does not satisfy this condition, what are the optimal set of capacities to remove so that the resulting network satisfies the condition?

The development of heuristics and algorithms for integer multicommodity flow problems remains an open research area. The "extended matroid graphs" of Chapter V were shown to provide an improvement for the MCTP. One should investigate their use in an algorithmic framework as proposed by Klingman and Russell [77] in solving the continuous

problem. In the algorithm for construction of EMG's, the capacity constraints on arcs incident to any node could be enforced. An interesting question is whether or not one can easily determine which set of constraints to enforce in order to provide the best lower bound, or to have a minimal amount of infeasibility in the resulting network.

It is hoped that this dissertation has laid the foundation for further unification and development of integer programming and networks by providing a fresh and unique characterization of certain integer programs, and a better understanding of the structure of multicommodity network flow problems.



## APPENDIX

## TUTTE'S ALGORITHM FOR GRAPHIC MATROIDS

The purpose of this appendix is to summarize, via an example, the terminology and steps of Tutte's algorithm for determining whether a binary matroid is graphic given in [128].

A binary chain group  $N$  on a finite set  $M$  is a class of subsets of  $M$  forming a group under mod 2 addition. These subsets are the chains of  $N$ . A chain is elementary if it is non-null and has no other non-null chain of  $N$  as a subset.

In our application, the set  $M$  will be the columns of an  $m \times n$  ( $m < n$ ) matrix of 0's and 1's. An example of a binary chain group is the class of all cuts of a linear graph,  $G$ . The corresponding matroid is called the bond-matroid of  $G$ . Tutte has developed necessary and sufficient conditions for a binary matroid to be graphic, that is, representable as the bond matroid of a graph.

The rank of a binary chain group  $N$  is the maximum number of linearly independent chains with respect to mod 2 addition. The structure of  $N$  is uniquely defined by a representative matrix  $R$  ( $m \times n$ ,  $m < n$ ). We will assume the rank of  $R$  to be  $m$ . The elements of  $R$  are 0's and 1's. The element in the  $i$ th row and  $j$ th column is 1 if the corresponding element of  $M$  (column  $j$ ) belongs to the corresponding chain of  $N$  (row  $i$ ). The chains of  $N$  are clearly linear combinations of the rows of  $R$ , the total number of chains being  $2^m$ . For example, let  $R$  be the following



$$R = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \end{matrix}$$

By elementary transformations of  $R$  including possibly a permutation of columns we can obtain a new representative matrix  $R_1$  of  $N$  in which the first  $m$  columns constitute a unit matrix.  $R_1$  is called a standard representative matrix of  $N$ . Tutte has shown that each row of  $R_1$  represents an elementary chain. For the example,  $R_1$  is

$$R_1 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{matrix}$$

Let  $\bar{M}$  be a binary matroid on the set  $M$ . The members of  $M$  are called cells of  $\bar{M}$ , and the members of the class  $\bar{M}$  (the "Q" in Axioms C1-C2) are called points by Tutte. (We have previously called them circuits). For example, in the matroid associated with the matrix  $R_1$  the point corresponding to the first row is the set  $\{1,6,7\}$ . Every row corresponds to a point.

We will now proceed with the algorithm for the example. As

new concepts or terminology arise, we will pause to define and illustrate them by the symbol  $\nabla$ .

Given a representative matrix  $R$  we transform it to a standard representative matrix  $R_1$ . If  $R_1$  contains at most two 1's in every column, then clearly  $\bar{M}$  is graphic for we need only take the mod 2 sum of rows and adjoin it to  $R_1$  to obtain a node-arc incidence matrix of a graph. Otherwise there is at least one column with three or more ones. Let us work with column 9. Let the first row with a 1 in column 9 correspond to a point  $Y$  of  $\bar{M}$ . Then  $Y = \{3, 6, 8, 9\}$  and  $Y \subseteq M$ .

$\nabla$  Let  $S \subseteq M$ . We define  $\bar{M} \cdot S$  as the set of points of  $\bar{M}$  which are minimal non-null intersections of  $S$  of points of  $\bar{M}$ . In other words,  $\bar{M} \cdot S$  consists of all points defined by the rows of  $R_1$  restricted to the columns  $\bar{M} \cap S$ .  $\bar{M} \cdot S$  is read " $\bar{M}$  restricted to  $S$ "  $\nabla$

The next step in the algorithm is to obtain the standard representative matrix  $R_2$  of  $\bar{M} \cdot (M - Y)$ . This can be obtained by deleting the row corresponding to  $Y$  and all columns having a one in that row, i.e., row 2 and columns 2, 8, 9, 10, leaving

$$R_2 = \begin{matrix} & \begin{matrix} 1 & 2 & 4 & 5 & 7 & 10 & 11 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \end{matrix}$$

$\nabla$  A separator of  $\bar{M}$  is a subset  $S$  such that no point of  $\bar{M}$  meets both  $S$  and  $M - S$ . A separator is elementary if it is non-null and

contains no other non-null separator. The elementary separators of  $M \cdot (\bar{M} - Y)$  are called the bridges of  $Y$  in  $M$ . ▼

The general rule for constructing an elementary separator is to take any row of  $R_2$ , then every row having a one in the same column as the first row chosen, then every row having a one in the same column as a row already chosen, and so on. The set of columns having a one in any of the rows of the resulting set is the separator. In  $R_2$  above the elementary separators are

$$B_1 = \{1, 4, 5, 7, 10, 11\}$$

$$B_2 = \{2\}$$

To each bridge  $B$  there corresponds a Y-component  $\bar{M}X(\text{BUY})$  of  $\bar{M}$ .

▼ If  $S \subseteq M$  then  $\bar{M}XS$  are those points of  $\bar{M}$  that are wholly contained in  $S$ . ▼

The Y-components are represented by submatrices of  $R_1$  whose rows correspond to a bridge with the row corresponding to  $Y$  adjoined to it. Thus, the Y component corresponding to  $B_1$  is represented by  $R_3$

$$R_3 = \begin{array}{c} \begin{array}{cccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{array} \\ \begin{array}{l} 1 \\ 4 \\ 5 \\ Y \rightarrow 3 \end{array} \left[ \begin{array}{cccccccccccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{array} \right] \end{array}$$

The Y-component corresponding to  $B_2$  is  $R_4$

$$R_4 = \begin{matrix} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ \begin{matrix} R_4 = 2 \\ Y \rightarrow 3 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \end{matrix}$$

▼ For each bridge  $B$  of  $Y$  in  $\bar{M}$ , the matroid  $(\bar{M}X(B \cup Y)) \cdot Y$  is of interest. It may happen that its points are disjoint subsets  $S_1, \dots, S_k$  of  $Y$  whose union is  $Y$ . If so, we say that  $B$  partitions  $Y$  and that  $\{S_1, \dots, S_k\}$  is the partition of  $Y$  determined by  $B$ . Then each standard representative matrix of  $(\bar{M}X(B \cup Y)) \cdot Y$  has one non-zero component in each column. ▼

By the definition of the operator  $\cdot$ , it is clear that  $(\bar{M}X(B_1 \cup Y)) \cdot Y$  is represented by all columns of  $R_3$  having a 1 in the row corresponding to  $Y$ , i.e.,

$$\begin{matrix} & 3 & 6 & 8 & 9 \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{matrix}$$

If this is reduced to standard form, we obtain

$$\begin{matrix} & 3 & 6 & 8 & 9 \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

This has only one 1 in each column. Hence,  $B_1$  partitions  $Y$  and the partition is  $\{\{16\}, \{8,9\}, \{3\}\}$ .  $B_2$  also partitions  $Y$ , yielding  $\{\{3,9\}, \{6,8\}\}$ . If the bridges had not partitioned  $Y$ , then  $\bar{M}$  is not graphic. We must now determine if  $Y$  is even.

▼ Let  $B$  and  $B'$  be bridges of  $Y$  in  $\bar{M}$  which partition  $Y$ , determining partitions  $\{S_1, \dots, S_k\}$  and  $\{T_1, \dots, T_m\}$  respectively. We call them non-overlapping bridges if there is an  $S_i$  and  $T_j$  such that  $Y = S_i \cup T_j$ . Otherwise  $B$  and  $B'$  overlap. We call  $Y$  an even point of  $\bar{M}$  if it satisfies

- (a) Each bridge of  $Y$  partitions  $Y$ .
- (b) The bridges of  $Y$  can be arranged into two disjoint classes so that no two members of the same class overlap. ▼

In the example, let the two disjoint classes be  $\{B_1\}$  and  $\{B_2\}$ . Clearly  $Y$  is even. We can now assert that  $\bar{M}$  is graphic if and only if the matroids corresponding to  $R_3$  and  $R_4$  are graphic. We now reapply the algorithm to  $R_3$  and  $R_4$ . But  $R_4$  contains at most two ones in each column, hence its matroid is graphic.

Consider  $R_3$ . We shall work with column 6 and let  $Y$  be row 1. Let  $M_3$  be the matroid corresponding to  $R_3$ . We first obtain  $M_3 \cdot (M - Y)$ .

$$\begin{array}{cccccccc} & 2 & 3 & 4 & 5 & 8 & 9 & 10 & 11 \\ \begin{array}{l} 4 \\ 5 \\ 3 \end{array} & \left[ \begin{array}{cccccccc} 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{array} \right] \end{array}$$

Then  $B_1 = \{2, 3, 4, 5, 8, 9, 10, 11\}$  and the Y-component is

$$R_4 = \begin{array}{c} \begin{array}{c} 4 \\ 5 \\ 3 \\ Y \rightarrow 1 \end{array} \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Then  $(M_3 \times (B_1 \cup Y)) \cdot Y$  is represented by

$$\begin{array}{c} \begin{array}{c} 1 & 6 & 7 \end{array} \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \end{array}$$

In standard form, this is

$$\begin{array}{c} \begin{array}{c} 1 & 6 & 7 \end{array} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \end{array}$$

Therefore  $B_1$  partitions  $Y$  and the partition is  $\{\{1\}, \{6\}, \{7\}\}$ .

Since there is only one bridge  $Y$  is clearly even. Hence  $M_3$  is graphic and therefore  $\bar{M}$  is graphic. If there was more than one bridge, we would have to reapply the algorithm again.

## BIBLIOGRAPHY

1. Arinal. J. C., "Maximal Biflow in an Undirected Network," IBM Journal of Research and Development, 13, 373-379, 1969.
2. Balas, E., "An Infeasibility Pricing Decomposition Method for Linear Programs," Opns. Res., 14, 847-873, 1966.
3. Balinski, M. L., "Integer Programming: Methods, Uses, Computation," Mgt. Sci., 12, 253-313, 1965.
4. Bazaraa, M. S., "An Infeasibility Pricing Algorithm for the Multicommodity Minimum Cost Flow Problem." Unpublished paper, School of Industrial and Systems Engineering, Georgia Institute of Technology, 1973.
5. Bazaraa, M. S. and J. J. Goode., "A Survey of Various Tactics for Generating Lagrangian Multipliers in the Context of Lagrangian Duality," Unpublished paper, School of Industrial and Systems Engineering, Georgia Institute of Technology, 1974.
6. Bellmore, M., H. J. Greenberg, and J. J. Jarvis, "Multicommodity Disconnecting Sets," Mgt. Sci., 16, B427-B433, 1970.
7. Bellmore, M. and R. Vemuganti, "On Multicommodity Maximal Dynamic Flows," Opns. Res., 21, 10-21, 1973.
8. Boyer, D. D., "A Modified Simplex Algorithm for Solving the Multicommodity Maximum Flow Problem," TM-14930, The George Washington University, Washington, D.C., 1968.
9. Bozoki, G., "Minimum Cost Multicommodity Network Flows," Ph.D. Dissertation, Purdue University, June, 1969.
10. Camion, P., "Characterization of Totally Unimodular Matrices," Proc. Amer. Math. Soc., 16, 1068-1073, 1965.
11. Cederbaum, I., "Matrices All of Whose Elements and Subdeterminants are 1, -1, or 0," J. Math and Phys., 36, 351-361, 1958.
12. Chandrascharan, R., "Total Unimodularity of Matrices," SIAM J. Appl. Math., 17, 1032-1034, 1968.
13. Chen, H. and C. G. DeWald, "A Generalized Chain Labelling Algorithm for Solving Large Multicommodity Flow Problems," Computers and Operations Research, 1, 437-466, 1975.

14. Commoner, F. G., "A Sufficient Condition for a Matrix to be Totally Unimodular," Networks, 3, 351-365, 1973.
15. Cremeans, J. E., R. A. Smith, and G. R. Tindall, "Optimal Multi-commodity Network Flows with Resource Allocation," Naval Res. Log. Quart., 17, 269-279, 1970.
16. Dantzig, G. B., "Applications of the Simplex Method to Transportation Problems," Activity Analysis of Production and Allocation, T. C. Koopmans, ed., Wiley, 1951.
17. Dantzig, G. B., Linear Programming and Extensions, Princeton University Press, 1963.
18. Desler, J. F. and S. L. Hakimi, "A Graph-Theoretic Approach to a Class of Integer Programming Problems," Opns. Res., 17, 1017-1033, 1969.
19. Edmonds, J., "An Introduction to Matching," University of Michigan Summer Conference, 1967.
20. Edmonds, J., "Matroids and the Greedy Algorithm," Math. Prog., 1, 127-136, 1971.
21. Edmonds, J., "Maximum Matching and a Polyhedron with 0-1 Vertices," J. Res. Nat. Bur. Std., 69B, 35-40, 1965.
22. Edmonds, J., "Minimum Partition of a Matroid Into Independent Subsets," J. Res. Nat. Bur. Std., 69B, 45-50, 1965.
23. Edmonds, J., "Paths, Trees, and Flowers," Can. J. Math., 17, 449-467, 1965.
24. Edmonds, J., "Submodular Functions, Matroids, and Certain Polyhedra," Combinatorial Structures and Their Applications, Gordon and Breach, 1970.
25. Edmonds, J. and D. R. Fulkerson, "Transversals and Matroid Partition," J. Res. Nat. Bur. Std., 69B, 57-63, 1965.
26. Fisher, M. L. and J. F. Shapiro, "Constructive Duality for Discrete Optimization," WP 625-72, Sloan School of Management, M.I.T., 1972.
27. Flood, M. M. "A Computational Algorithm for the Hitchcock Distribution Problem," University of Michigan, 1959.
28. Ford, L. R. and D. R. Fulkerson, "A Primal-Dual Algorithm for the Capacitated Hitchcock Problem," Naval Res. Log. Quart., 4, 47-54, 1957.



29. Ford, L. R. and D. R. Fulkerson, "A Suggested Computation for Maximal Multicommodity Network Flows," Man. Sci., 5, 97-101, 1958.
30. Ford, L. R. and D. R. Fulkerson, Flows in Networks, Princeton University Press, 1962.
31. Ford, L. R. and D. R. Fulkerson, "Maximal Flow Through a Network," Can. J. Math., 8, 399-404, 1956.
32. Ford, L. R. and D. R. Fulkerson, "Solving the Transportation Problem," Man. Sci., 3, 24-32, 1956.
33. Frank, H. and I. T. Frisch, Communication, Transmission, and Transportation Networks, Addison-Wesley, 1971.
34. Garfinkel, R. S. and G. L. Nemhauser, Integer Programming, Wiley, 1972.
35. Glover, F., "Maximum Matching in a Convex Bipartite Graph," Naval Res. Log. Quart., 14, 313-316, 1967.
36. Glover, F., D. Karney, and D. Klingman, "Computational Comparisons of Primal, Dual, and Primal-Dual Computer Codes for Minimum Cost Network Flow Problems," Networks, 4, 191-212, 1974.
37. Glover, F., D. Karney, D. Klingman, and A. Napier, "A Computation Study on Start Procedures, Basis Change Criteria, and Solution Algorithms for Transportation Problems," Man. Sci., 20, 793-813, 1974.
38. Grigoriadis, M. D. and W. W. White, "A Partitioning Algorithm for the Multicommodity Network Flow Problem," Math. Prog., 3, 157-177, 1972.
39. Grigoriadis, M. D. and W. W. White, "Computational Experience with a Multicommodity Network Flow Algorithm" in Optimization Methods for Resource Allocation, R. Cottle and J. Krarup, eds., English University Press, 1972.
40. Grinold, R. C., "A Multicommodity Max-Flow Algorithm," Opns. Res., 16, 1234-1238, 1968.
41. Grinold, R. C., "A Note on Multicommodity Max Flow," Opns. Res., 17, 755, 1969.
42. Hadley, G., Linear Programming, Addison-Wesley, 1962.
43. Hakimi, S. L., "On Simultaneous Flows in a Communication Network," IRE Trans., CT-9, 1962.

44. Harary, F. and D. Welsh, "Matroids versus Graphs" in The Many Facets of Graph Theory, G. Chartrand and S. F. Kapoor, eds., Springer-Verlag, 1969.
45. Hartman, J. K. and L. S. Lasdon, "A Generalized Upper Bounding Algorithm for Multicommodity Network Flow Problems," Networks, 1, 333-354, 1972.
46. Held, M., P. Wolfe, and H. Crowder, "Validation of Subgradient Optimization," Math. Prog., 6, 62-88, 1974.
47. Heller, I., "Constraint Matrices of Transportation-Type Problems," Naval Res. Log. Quart., 4, 73-78, 1957.
48. Heller, I., "On a Class of Equivalent Systems of Linear Inequalities," Pac. J. Math., 13, 1209-1227, 1963.
49. Heller, I., "On Linear Programs Equivalent to the Transportation Problem," SIAM J. Appl. Math., 14, 31-41, 1964.
50. Heller, I., "On Linear Systems with Integral Valued Solutions," Pac. J. Math., 7, 1351-1364, 1957.
51. Heller, I., "On Unimodular Sets of Vectors," in Recent Advances in Mathematical Programming, R. Graves and P. Wolfe, eds., McGraw-Hill, 1963.
52. Heller, I. and A. J. Hoffman, "On Unimodular Matrices," Pac. J. Math., 12, 1321-1326, 1962.
53. Heller, I. and C. B. Tompkins, "An Extension of a Theorem of Dantzig's," in Linear Inequalities and Related Systems, H. Kuhn and A. W. Tucker, eds., Princeton University Press, 1956.
54. Hitchcock, F. L., "The Distribution of a Product from Sources to Numerous Localities," Math. Phys., 20, 223-230, 1941.
55. Hoffman, A. J. and J. B. Kruskal, "Integral Boundary Points of Convex Polyhedra," in Linear Inequalities and Related Systems, H. Kuhn and A. W. Tucker, eds., Princeton University Press, 1956.
56. Hu, T. C., Integer Programming and Network Flows, Addison-Wesley, 1969.
57. Hu, T. C., "Multicommodity Network Flows," Opns. Res., 11, 344-360, 1963.
58. Hu, T. C., "On the Feasibility of Multicommodity Flows in a Network," Opns. Res., 12, 359-360, 1964.

59. Ingleton, A. W., "Representation of Matroids," in Combinatorial Mathematics and Its Applications, D. J. A. Welsh, ed., Academic Press, 1971.
60. Iri, M., "On An Extension of the Maximum-Flow Minimum-Cut Theorem to Multicommodity Networks," J. Opns. Res. Soc. Japan, 13, 1971.
61. Jarvis, J. J., "On the Equivalence Between the Node-Arc and Arc-Chain Formulations for the Multicommodity Maximal Flow Problem," Naval Res. Log. Quart., 16, 515-529, 1969.
62. Jarvis, John J. Private Communication.
63. Jarvis, J. J. and P. K. Keith, "Multicommodity Flows with Upper and Lower Bounds," Unpublished paper, School of Industrial and Systems Engineering, Georgia Institute of Technology, 1974.
64. Jarvis, J. J. and J. B. Tindall, "Disconnecting Multicommodity Networks Through Primal Labelling," Unpublished paper, School of Industrial and Systems Engineering, Georgia Institute of Technology.
65. Jarvis, J. J. and J. B. Tindall, "Minimal Disconnecting Sets in Directed Multi-Commodity Networks," Naval Res. Log. Quart., 19, 681-690, 1972.
66. Jewell, W. S., "A Primal-Dual Multicommodity Flow Algorithm," ORC66-24, University of California, Berkeley, 1966.
67. Jewell, W. S., "Multicommodity Network Solutions," in Theorie des Graphes, Actes des Journees Internationales d'Etudes de l'I.C.C.-Rome, 1966, Donod, Paris, 1967.
68. Jewell, W. S., "Optimal Flow Through Networks," Interim Technical Report No. 8., M.I.T. Project, Fundamental Investigations in Methods of Operations Research, 1958.
69. Johnson, E. L., "Networks and Basic Solutions," Opns. Res., 14, 619-623, 1969.
70. Johnson, E. L., "Programming in Networks and Graphs," ORC 65-1, University of California, Berkeley, 1965.
71. Kantorovich, L. U. and M. K. Gavurin, "The Application of Mathematical Methods to Problems of Freight Flow Analysis," Adadomii Nauk SSR, 1949.
72. Karp, R. M., "Reducibility Among Combinatorial Problems," Complexity of Computer Computations, Plenum Press, 1972.

73. Keith, P. K., "A Solution Procedure for the Strategic Transportation Problem," M.S. Thesis, Georgia Institute of Technology, 1973.
74. Kennington, J. L., "Multicommodity Network Flows: A Survey," Technical Report CP74015, Department of Computer Science and Operations Research, Southern Methodist University, 1974.
75. Kleitman, D. J., "An Algorithm for Certain Multi-Commodity Flow Problems," Networks, 1, 75-90, 1971.
76. Kleitman, D., A. Martin-Löf, B. Rothschild, and A. Whinston, "A Matching Theorem for Graphs," J. Comb. Theory, 8, 104-114, 1970.
77. Klingman, D. and R. Russell, "Solving Constrained Transportation Problems," Opns. Res., 23, 91-106, 1975.
78. Kuhn, H. W., "The Hungarian Method for the Assignment Problem," Naval Res. Log. Quart., 2, 83-97, 1955.
79. Langley, R. W., "Continuous and Integer Generalized Flow Problems," Ph.D. Dissertation, Georgia Institute of Technology, 1973.
80. Langley, R. W. and J. L. Kennington, "The Transportation Problem: A Primal Approach," Presented to the 43rd ORSA National Meeting, May, 1973.
81. Lasdon, L. S., Optimization Theory for Large Systems, MacMillan, 1970.
82. Lawler, E. L., "Matroid Intersection Algorithms," ERL-M333, Electronics Research Laboratory, University of California, Berkeley, 1971.
83. Lawler, E. L., "Matroids with Parity Conditions: A New Class of Combinatorial Optimization Problems," ERL-M334, Electronics Research Laboratory, University of California, Berkeley, 1971.
84. Lawler, E. L., "Polynomial Bounded and (Apparently) Non-Polynomial Bounded Matroid Computations," ERL-M332, Electronics Research Laboratory, University of California, Berkeley, 1972.
85. Liu, C. L., Introduction to Combinatorial Mathematics, McGraw-Hill, 1968.
86. MacLane, S., "Some Interpretations of Abstract Linear Dependence in Terms of Projective Geometry," Amer. J. Math., 58, 236-240, 1936.
87. Maier, S., "A Multi-Item Network Flow Model with Capacity Constraints," Technical Report 69-14, O.R. House, Stanford University, 1969.

88. Markland, R. E., "Analyzing Multi-Commodity Production-Distribution Networks with Transformed Flows," School of Business Administration, University of Missouri-St. Louis, 1972.
89. McLoughlin, G. T., "Transportation Networks," Presented to the 44th National ORSA Meeting, San Diego, 1973.
90. Minty, G. J., "On the Axiomatic Foundations of the Theories of Directed Linear Graphs, Electrical Networks, and Network Programming," J. Math. Mech., 15, 485-520, 1966.
91. Morrison, D. R., "Matching Algorithms," J. Comb. Theory, 6, 20-32, 1969.
92. Munkres, J., "Algorithms for the Assignment and Transportation Problems," SIAM J. Appl. Math., 5, 32-38, 1957.
93. Naniwada, M., "Multicommodity Flows in a Communication Network," Elect. Comm. Japan, 52-A, 34-40, 1969.
94. Onaga, K., "A Multicommodity Flow Theorem," Elect. Comm. Japan, 53-A, 16-22, 1970.
95. Pierskalla, W. P., "The Multidimensional Assignment Problem," Opns. Res., 16, 422-431, 1968.
96. Piff, M. J. and D. J. A. Welsh, "On the Vector Representation of Matroids," J. London Math. Soc., 2, 284-288, 1970.
97. Ponstein, J. A., "A Generalization of the Incidence Matrices of a Graph," Theory of Graphs, International Symposium, Rome 1966, Dunod, Paris, 1968. ✓
98. Potts, R. B. and R. M. Oliver, Flows in Transportation Networks, Academic Press, 1972.
99. Rado, R., "A Theorem on Independence Relations," Quart. J. Math., 13, 83-89, 1942.
100. Reban, K. R., "Total Unimodularity and the Transportation Problem: A Generalization," Linear Algebra and Its Applications, 8, 11-24, 1974.
101. Robacker, J. T., "Concerning Multicommodity Networks," RM-1799, The Rand Corp., Santa Monica, California, 1956.
102. Rockafellar, R. T., "The Elementary Vectors of a Subspace of  $R^N$ ," Combinatorial Mathematics and Its Applications, University of North Carolina Press, 1969.
103. Rosen, J. B., "Primal Partition Programming for Block Diagonal Matrices," Num. Math., 6, 250-260, 1964.



104. Rothfarb, B. and I. T. Frisch, "Common Terminal Multicommodity Flow with a Concave Objective Function," SIAM J. Appl. Math., 18, 485-503, 1970.
105. Rothfarb, B. and I. T. Frisch, "On the Three-Commodity Flow Problem," SIAM J. Appl. Math., 17, 46-58, 1969.
106. Rothfarb, B., N. P. Schein, and I. T. Frisch, "Common Terminal Multicommodity Flow," Opns. Res., 16, 202-204, 1968.
107. Rothschild, B. and A. Whinston, "Feasibility of Two-Commodity Network Flows," Opns. Res., 14, 1121-1129, 1966.
108. Rothschild, B. and A. Whinston, "Maximal Two-Way Flows," SIAM J. Appl. Math., 15, 1967.
109. Rothschild, B. and A. Whinston, "Multicommodity Network Flows," Unpublished Draft, (see References 107 and 108), 1969.
110. Rothschild, B. and A. Whinston, "Multicommodity Network Flows with Multiple Sources and Sinks," Paper 193, Institute for Research in the Behavioral, Economic, and Management Sciences, Krannert School of Industrial Administration, Purdue University, 1967.
111. Rothschild, B. and A. Whinston, "On Two-Commodity Network Flows," Opns. Res., 14, 377-388, 1966.
112. Rothschild, B., A. Whinston, and J. Kent, "Computing Two Commodity Flows," Opns. Res., 16, 446, 1968.
113. Saigal, R., "Multicommodity Flows in Directed Networks," ORC 67-38, University of California, Berkeley, 1967.
114. Saigal, R., "On the Modularity of a Matrix," Linear Algebra and its Applications, 5, 39-48, 1972.
115. Sakarovitch, M., "The Multicommodity Maximum Flow Problem," ORC 66-25, University of California, Berkeley, 1966.
116. Seshu, S. and M. Reed, Linear Graphs and Electrical Networks, Addison-Wesley, 1961.
117. Srinivasan, V. and G. L. Thompson, "Accelerated Algorithms for Labelling and Relabelling of Trees with Applications to Distribution Problems," J. ACM, 19, 712-726, 1972.
118. Swoveland, C., "Decomposition Algorithms for the Multicommodity Distribution Problem," Working Paper 184, Western Management Science Institute, University of California, Los Angeles, 1971.

119. Tang, D. T., "Bipath Networks and Multicommodity Flows," IEEE Trans., CT-11, 468-474, 1964.
120. Tang, D. T., "Comments on Feasibility Conditions of Simultaneous Flows in a Network," Opns. Res., 13, 176-182, 1965.
121. Tang, D. T., "Communication Networks with Simultaneous Flow Requirements," IRE Trans. on Circuit Theory, CT-9, 176-182, 1962.
122. Tapiero, C. S. and M. K. Starr, "Assignment of N Products to M Markets with Various Interdependent Constraints Using a Graph-Boolean Approach," Columbia University Graduate School of Business, 1970.
123. Tomizawa, N., "On Some Techniques Useful for Solution of Transportation Network Problems," Networks, 1, 173-194, 1971.
124. Tomlin, J. A., "Minimum Cost Multicommodity Network Flows," Opns. Res., 14, 45-51, 1966.
125. Tucker, A. W., "A Combinatorial Equivalence of Matrices," Proceedings of Symposia in Applied Mathematics, Vol. 10: Combinatorial Analysis, R. Bellman and M. Hall, eds., 129-140, 1960.
126. Tutte, W. T., "A Class of Abelian Groups," Can. J. Math., 8, 13-28, 1956.
127. Tutte, W. T., "A Homotopy Theory for Matroids, I, II," Trans. A.M.S., 88, 144-174, 1958.
128. Tutte, W. T., "An Algorithm for Determining Whether a Given Binary Matroid is Graphic," Proc. A.M.S., 11, 905-917, 1960.
129. Tutte, W. T., "From Matrices to Graphs," Can. J. Math., 16, 108-127, 1964.
130. Tutte, W. T., Introduction to the Theory of Matroids, American Elsevier, 1971.
131. Tutte, W. T., "Lectures on Matroids," J. Res. Nat. Bur. Std., 69B, 1-47, 1965.
132. Tutte, W. T., "Matroids and Graphs," Trans. A.M.S., 90, 527-552, 1959.
133. Weigel, H. S. and J. E. Cremeans, "The Multicommodity Network Flow Model Revised to Include Vehicle Per Time Period and Node Constraints," Naval Res. Log. Quart., 19, 77-90, 1972.
134. Welsh, D. J. A., "Euler and Bipartite Matroids," J. Comb. Theory, 6, 375-377, 1969.

135. Whitney, H., "On the Abstract Properties of Linear Dependence," Amer. J. Math., 57, 509-533, 1935.
136. Williamson, J., "Determinants Whose Elements are 0 and 1," Amer. Math. Monthly, 53, 427-434, 1946.
137. Wilson, R. J., "An Introduction to Matroid Theory," Amer. Math. Monthly, 80, 500-525, 1973.
138. Wollmer, R. D., "Multicommodity Networks with Resource Constraints: The General Multicommodity Flow Problem," TM393-50, Jet Propulsion Laboratory, California Institute of Technology, 1971.
139. Wollmer, R. D., "The Dantzig-Wolfe Decomposition Principle and Minimum Cost Multicommodity Network Flows," P-4191, The Rand Corporation, Santa Monica, California, 1969.
140. Yatsuboshi, R., "On Feasibility Conditions of Multicommodity Flows in a Network," Elect. Comm. Japan, 55-A, 1-7, 1972.



## VITA

James R. Evans was born on March 26, 1950 in Chicago, Illinois. He graduated from St. Ignatius College Prep in 1968 and immediately entered Purdue University. Mr. Evans received a Bachelor of Science in Industrial Engineering "with highest distinction" in January, 1972, and a Master of Science in Industrial Engineering in June, 1972. At Purdue he was elected to Phi Eta Sigma, Alpha Pi Mu, Tau Beta Pi, Omicron Delta Kappa, and Phi Kappa Phi; and he received the 1970 H. T. Amrine Award in Industrial Engineering and the 1972 Armstrong Cork Co. Student Award for Excellence from the American Institute of Industrial Engineers.

During the summers, Mr. Evans was employed as an analyst by the U.S. Army Computer Systems Command and subsequently as an industrial engineer for the Chicago, Rock Island, and Pacific Railroad.

He entered Georgia Tech in 1972 to pursue a Ph.D. in Operations Research in the School of Industrial and Systems Engineering. There he was elected to Pi Mu Epsilon and received a 1974 Honorable Mention in the ORSA Student Paper Competition. He received his Ph.D. in 1975 and accepted a position as assistant professor in the Department of Quantitative Analysis at the University of Cincinnati.